ASIRI, MAHA MOHAMMED, M.S, A Transfer Learning-Based Feature Reduction Method To Improve Classification Accuracy. (2017)
Directed by Dr. Fereidoon Sadri and Dr. Hamid Nemati. 166 pp

The need for efficient data use grows in machine learning algorithm for dataset with larger feature sets. Feature selection is the process of selecting minimum set of features that fully represent the learning problem. Transfer learning can motivate in scenario where we train model with the common problem and use it to identify important features needed to build model for target problem.

In this thesis, we propose transfer learning algorithm combined with or without suggested features from experts, to learn from the source dataset and recognize important feature sets needed to train models in target dataset. Also, we compared this algorithm with classical machine learning algorithm with or without using the suggested features recommended by the experts. In series of experiment, it shows that our method is adequate to find the minimum feature sets which also outperformed then using only the suggested features by the experts. Furthermore, it also shows that the subsequent reduce in number of features in transfer learning method have better or almost same performance then using all the features of the dataset. We performed our experiments using heart disease, readmission dataset and BMI dataset.

# A TRANSFER LEARNING-BASED FEATURE REDUCTION METHOD TO IMPROVE CLASSIFICATION ACCURACY

by

Maha Mohammed Asiri

A Thesis Submitted to
the Faculty of The Graduate School at
The University of North Carolina at Greensboro
in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Greensboro
2017

Approved by

_____
Committee Chair

APPROVAL PAGE

This thesis written by MAHA MOHAMMED ASIRI has been approved by the following committee of the Faculty of the Graduate School at The University of North Carolina at Greensboro.

Committee Chair_____
Fereidoon Sadri

Committee Members_____
Hamid Nemati

_____
Nancy Green

November 14, 2017
Date of Acceptance by Committee

November 14, 2017
Date of Final Oral Examination

## ACKNOWLEDGMENTS

Dedicated to my wonderful parents, Mohammed and Khadijah, my dearest sisters, Reem, Shahad, Ahad, Raghad, Majd and Jood, my amazing brother Abdullah, and my sweet niece Layan.

All thanks and praise be to Allah for letting me through all the difficulties and helping me to complete my master's degree successfully.

I would like to thank my parents for their love. Thanks for supporting me during my studies and urging me on. Thank you both for giving me strength to reach for the stars and chase my dreams. Words cannot express how grateful I am for all of the sacrifices that you've made on my behalf. Your prayer for me was what sustained me thus far. Mom and Dad, you are wonderful parents and wonderful friends. Also, I am extremely grateful to my sisters, brother and niece for their love, understanding, prayers and continuing support to complete this research work.

I would like to express my special appreciation and thanks to my advisor Professor Dr. Sadri for his guidance and support throughout this study, you have been a tremendous mentor for me. I truly appreciated all the time and advice you gave me throughout my time at UNCG. Finally, thanks for treating me with respect and timely advice during my graduate studies.

Thank you to Dr. Hamid Nemati for acting as a second advisor to me. Thanks for taking the time to teach me what you know and helping me throughout my graduate studies. It was a great privilege and honor to work and study under your guidance.

I would also like to acknowledge my committee member Dr.Nancy Green, who agreed to serve on my committee. I appreciated your guidance, support and willingness to take time to discuss my research.

A special thanks to my sister, Dr.Shahad Asiri, who participated in this research and provided me assistance on understanding the medical data by sharing her expertise in the medical field. Thank you for making me so proud of you.

I am also grateful for the support and encouragement of Sudip Lama who shared his great knowledge with me. Without his help, brilliant comments and suggestions, I could not have completed this thesis.

To all my friends who have supported me to complete the research work directly or indirectly, thank you for your understanding and encouragement. Your friendship makes my life a wonderful experience. I cannot list all the names here, but you are always on my mind.

Last but certainly not least, thanks to the government of my country and King Khalid University. I could not have gone through the master program overseas without their financial support.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

# CHAPTER I

# INTRODUCTION

## 1.1 Introduction

The field of data mining and machine learning has been widely and successfully used in many applications where patterns can be extracted from past information (training data) to predict future outcomes [4]. Machine learning has its advantages in all walks of life, with applications ranging from autonomous cars [7], stock value prediction [12], heart disease prediction [9] and even cancer diagnosis [11, 8].

Usually, the data is described by a set of features. We call features unnecessary if they are either irrelevant to the current goal or hold redundant information given other features. Many machine learning algorithms tend to get overwhelmed when unnecessary data abounds. They usually need more samples in the presence of irrelevant features. For example, the number of training samples needed for the basic nearest-neighbor classification algorithm to reach a given accuracy grows exponentially with the number of irrelevant features [10]. However, the success of supervised learning techniques depends on the presence of sufficiently large sets of training data. Ideally, these training sets are sampled from the same generating distribution that is expected to be present in production. Obtaining useful training sets is most often an arduous and expensive process. Transfer learning techniques allow us to reuse knowledge (such as models or examples) gained from some learning task (called the source) and apply it to a related task for which enough training sets are not yet available (called the target). Effective transfer learning techniques are much in need due to the growing demand for machine learning solutions for an ever-increasing number of computer applications and the tremendous growth in communicated information.

Consider, for example, the problem of automatic heart disease prediction using health tracking mobile app. The proportion of heart disease records to legitimate ones is quite small in the context of any single mobile app user. Thus, the corresponding learning problem can be viewed as binary classification with a tiny minority target class. Consequently, the acquisition of a sufficiently large labeled training set may take considerable time. If we already possess an annotated database of heart disease patient records from several other users or hospital (‘source’), we could, hypothetically, use it for the current challenge (‘target’).

To address the problem above the transfer learning attempt to utilize whatever source information is available, guided by the spares information already acquired for the new target.

Our work focuses on inductive transfer learning, a setting in which one assumes that both source and target tasks share the same features and label spaces. Most work in the inductive transfer learning setting has focused on meta-algorithms, algorithms that operate on existing machine learning techniques [14, 15, 16, 17, 18, 19, 20, 21, 22]. In contrast, our work takes the popular and well-researched approach of decision trees and applies it to the inductive transfer learning problem. The rest of this thesis is organized as follows. In Chapter 2, we present the theoretical background on machine learning and transfer learning. We also discuss some of the commonly adopted algorithms and techniques used in machine learning and transfer learning. In Chapter 3, we discuss about the architectures and algorithms followed during the experiments. Later, in Chapter 4, we discuss the dataset used to for the experiment. We also provided empirical evidence that shows the advantage of our proposed algorithms and comparison between different techniques. Our final observations and still open questions are provided with the concluding remarks in Chapter 5.

# CHAPTER II

# LITERATURE REVIEW

## 2.1 Machine Learning

Machine learning is an artificial intelligence (AI) which gives room for software applications to turn out to be more accurate in forecasting. Machine learning is a technique of data analysis that utilizes algorithms that acquire information and learn from data and come up with definite results without the need to be programmed specifically to do so [86]. The algorithms have the ability to analyze data, calculate the frequency of how certain parts of the data are used and consequently come up with responses grounded on the calculations and deductions so as to be able to automatically interact with the users.

Machine learning has been employed in many platforms in the world today ranging from the generation of the 'other items you may be interested in' responses at sites like Amazon to the provision of avenues to detect fraud, generation of web search results and the filtration of spam in e-mail servers. It is also being used in medicine especially in medical data provision [68]. Machine learning offers the users a chance to identify patterns or dependencies which may not be visible to a human.

Object clustering procedures provide room for grouping huge quantity of data by the use of wide range of meaningful criteria. Humans cannot operate efficiently with numerous objects exceeding hundreds of entities with many structures. On the other hand, machines (computers) are able to perform clustering with much efficiency, for instance, customer/ leads qualification, product lists segmentation. Recommendation/ preferences/ behavior prediction algorithms offer room for more efficiency in the interaction of clients or users by giving them what they require [83].

Recommendation systems have some problems in working in some services but it is expected to improve with time.

**2.2 Machine Learning in Medical Data**

The aim of machine learning is to provide methods of computational, varying and revising information in intellectual systems and especially in education systems that assist in the induction of understanding from cases or data [82]. Methods in machine learning are helpful in situations where algorithmic resolutions are not accessible, there is no official representation or the know-how on the appliance sphere is inadequately defined.

In a medical application, machine learning gives the processes, procedures, and paraphernalia that can give solutions for problem-solving and extrapolative tribulations in an assortment of medical areas. Machine learning is employed in the medicine especially in the analysis of the significance of medical strictures and of their permutation for prognosis. For example, forecasting of infection development for the mining of information in medicine, especially for the exploration of outcomes can be done by machine learning. Machine learning is also useful in the analysis of data, for example, the detention of data promptness in the facts through appropriation dealing with flawed statistics, the elucidation of incessant statistics employed in the intensive care unit (ICU) and for intellectual disquieting causing to operative and competent observation. With proper implementation, machine learning methods can be employed in easing the incorporation of computer-based systems in the healthcare field giving chances to improve the medical expert's job and consequently progress the effectiveness and eminence of medical care.

Reasoning in medical diagnostic is crucial in the fields of computer-based systems. In this field, specialist scheme and the model-based system give room for a device for the coming up with theories extracted from patient facts. For instance, regulations are dug up from the information of professionals in the specialist scheme. The problem is that on most of the cases the professionals

or experts may not have the information or may not be able to come up with the formula of the knowledge required in the solving of the problems experienced. By the use of figurative learning procedures (for example inductive learning by examples) are employed to enhance education and information supervision competencies to specialists' schemes. For instance, when accorded with a set of clinical scenarios as an example, the features that are clinical and uniquely give the traits of the clinical situation. This data can be laid out in the shape of uncomplicated regulations or as a decision tree. An example of this presentation of the system is KARDIO which was a brainchild to give an interpretation of ECGs.

This kind of approach can be protracted to take care of scenarios where previous experience is not available in the understanding and indulgence of medicinal statistics. For instance in the drudgery of Hau and Coicera, a system that is intelligent, that can take a concurrent patient information extracted through a cardiac bypass surgery and consequently employed in the creation of representation of typical and anomalous cardiac functioning for the recognition of vicissitudes in the conditions of a patient is portrayed [82]. In addition, in a research setting, these models can be employed to provide the initial hypotheses that can build and initiate the need for further experimentation.

In the process of learning from the data obtained from patients faces some difficulties due to the fact that in most cases the data sets have attributes of incompleteness (omitted values of some parameters) erroneousness (methodical or arbitrary racket in the information), sparseness (insufficient and/or non-representable patient accounts accessible) a vagueness (incongruous miscellany of bounds for the given task). Machine learning accords the paraphernalia for dealing with these traits of the medicinal datasets. Subsymbolic techniques of learning for instance in the neural grid are competent of handling the datasets and they are mostly employed due to their

matching of designs aptitudes and their possession of human-like physiognomies (simplification heftiness to noise) in the feat to advance the health check verdict making criteria.

Another field where machine learning is functional in medicine is biomedical processing. The perception of biological structure is not so much known and it is incomplete but there are features and facts concealed in the physiological systems which are not enthusiastically ostensible. In other hands, the paraphernalia between various subsystems is not discernible. The characterization of the biological signatures is through substantial variability brought about by either impulsive mechanisms that are internal or external stimuli [82]. The association between the various traits can be much complex to provide any solution using techniques that are conservative. Machine learning techniques much depend on these sets of data which can be fashioned easily and can bring about assistance in the modeling of non-linear associations that are existent between these data and excerpt considerations and features that can be used to progress medical care.

The PC base medical imaging system involves the use of area giving important support in the analysis of medicine. In many scenarios expansion of the systems is deliberated as an effort to contend with the doctor's proficiency. This is the proof of identity of malevolent areas in triflingly intrusive imaging techniques. For example, there is computed tomography, ultrasonography, endoscopy, confocal microscopy, computed radiography or magnetic resonance imaging. The main idea is the increase of aptitude of professionals in the field of cancer diagnosis regions while doing away with the urge for intrusion and maintenance of the capability for precise analysis. In addition, there is the possibility of the examination of bigger areas, the study of living tissues in vivo, most probably from a distance and therefore decrease the inadequacies of biopsies like the uneasiness of the patients, diagnosis delay and the restricted number of tissue samples. The requirement of prompt discovery like those that the computer-aided medical analysis systems aspire to present is imminent.

In essence, the healthcare industry has evolved to become more and more dependent on technology brought about by the increased use of computers. By the application of machine learning methods, there is the possibility of provision of helpful aids that can give assistance to physicians in numerous occasions. This can also help in doing away with issues and problems which are associated with the fatigue of human beings, help in the provision of speedy identification of abnormalities and problems and help in the enabling of making a real-time diagnosis of patients.

**2.3 Algorithms of Machine Learning**

Machine learning which is an artificial intelligence that learns on itself through the identification of new patterns in data allows scientist and other users to efficiently pinpoint revenue opportunities and come up with approaches to advance the experiences of clients by the exploitation of the information usually hidden in enormous sets of data. There are three broad types of machine algorithms [84]. They include supervised learning. This type of algorithm entails of an objective (reliant variable) that is likely to be forecasted from a certain set of predictors (independent variables). By the exploitation of the variables generation of functions that can map inputs to the anticipated outcomes. The process of training takes place until the representation attains a sought after echelon of accurateness on the instruction data. Examples of supervised learning include regression, decision tree, random forest, KNN, and Logistic Regression among others.

Unsupervised Learning is an algorithm that does not involve any objective or result variable to be predicted or estimated. It is employed in the clustering of populations in various groups mostly employed in the segmentation of clients into diverse clusters with specific interventions. Examples include Apriori algorithm and K-means [77].

Reinforcement Learning involves the contraption being taught to create exact conclusions. The working of this algorithm involves the machine is open to the elements of a milieu where it guides itself recurrently by exploiting trial and error method. The machine mugs up from the

proficiencies from the past and it attempts to internment the finest probable information to construct precise verdicts [73]. Examples include Markov Decision Process.

The selection of the right algorithm is an essential key in every part of any machine learning project especially with the dozens of algorithms on offer. Understanding weaknesses and strengths of the different algorithms in their application is also important. The most common machine learning algorithms and their potential use cases include random forest, neural networks, and decision trees.

### 2.3.1 Decision Tree Classification Algorithm

Decision trees are powerful and most common tools for the classification and prediction of data. Decision trees are used in the representation of rules that are understandable to humans and employed in the knowledge systems like the databases. The key requirements of decision tree classification algorithm include attribute-value descriptions. These are the objects or cases that should be expressed in terms or forms of fixed collection of materials or attributes. For instance, there is hot, mild or cold [78]. Another requirement id predefined classes or the target values. These target functions contain isolated output values, for example, boolean or multiclass. There is also sufficient data which in most cases enough training scenarios should be given and presented in the leaning of the given model.

The decision tree helps in the building of classifications or regressions models that are in the form of tree structures. In its working, it breaks down a dataset into minute subsets and in the same time, it develops an associated decision tree that is more developed. The expected results of the tree have decision nodes and leaf nodes [75]. Decision nodes like the outlook consist of more than two branches. The leaf node gives out a classification or decisions. The uppermost decisions in the tree correspond to the finest predictor which is referred to as root node. Decision trees are attributed with the ability to handle both definite and arithmetical data.

The algorithm that is used in the construction of decision tree in this study is the CART. The CART algorithm is prearranged as the progression of questions, the respond of whom determine what the next question should be [74]. The results derived from these questions are structured like trees in which the ends are the terminal nodes in which extra questions are not applicable. The major rudiments of CART (and any other decision tree algorithm) include: the laws for the splitting of data at a node based on the value of the variable, the stopping rules for choosing when a branch is terminal and no more splitting can be done and the prediction for the target variable in each of the terminal node.

Decision tree classifier has a set of parameters with different values that can be modified. The first parameter is criterion, which is the function to measure the quality of the split in the tree. In scikit-learn it has the default value "gini" for the Gini impurity. Below is the formula of Gini:

$$\text{Gini: Gini(E)}=1-\sum_{j=1}^{c} p_j^2$$

The second parameter is the splitter which is the strategy that is used to choose the split at each node, it has the default value "best" to choose the best split. Another parameter is "max_depth" which is the maximum depth of the tree. By default, it has the value "None" which means that nodes are expand until all leaves are pure. However, in our study we changed the value of the maximum depth to be from 1 to 20. Decision tree classifier also has the parameter "min_samples_split" which is the minimum number of samples required to split an internal node, it has the default value 2. Another parameter is "min_samples_leaf" which is the minimum number of samples required to be at a leaf node, it has the default value 1. "Min_weight_fraction_leaf" is one of the parameters that means the minimum weighted fraction of the sum total of weights required to be at a leaf node, it has the default value 0. When sample_weight is not provided then,

samples have equal weight. The number of features to consider when looking for the best split is also one of the classifier parameter “max_features”, its default value is “None”. That means the maximum number of features is equal to the total number of the features. Another parameter is “max_leaf_nodes” which has the default value “None”, it means grow a tree with unlimited number of leaf nodes. “Class_weight” is one of the parameter which means the weights associated with the classes and it has the default value “None”. Decision tree classifier also has the parameter “presort” which is Boolean and by default is false. It is used to presort the data to speed up the finding of best splits in fitting. For the settings of a decision tree on a small dataset or a restricted depth, setting this to true may speed up the training. On the other hand, when using a large dataset, this may slow down the training process [91].

For our experiments, we had created 20 different models of decision tree in two different ways.

- Create models by varying max_depth from 1 to 20.
- Create models by varying max_depth from 4 to 8 and min_samples_split from 20 to 50.

### 2.3.2 Random Forest Algorithm

Random forest algorithm is a supervised classification algorithm that creates the forest with various trees. The more the number of trees in the forest the more robust the forest is expected to look like and the higher then numbers of trees in a given forest the higher the results are accurate. The random forests are a mishmash of tree predictors like that individual tree rely on the value of a random vector usually tested separately and in the similar allocation for all the trees in the forest.

In random forest, decision trees use directed graphs to model decision making with each node on the graph representing a question concerning the data (“Is income greater than $60,000) and the branches which sprout from individual nodes denote the possible clues to the question [71].

The compounding of hundreds or thousands of decisions trees is an ensemble method referred to as random forest.

Random forest classifier has a set of parameters with different values that can be modified. The first parameter is the number of trees in the forest “n_estimators”, the default value is 10. The second parameter is criterion, which is the function to measure the quality of the split in the tree. Same as the decision tree classifier, it has the default value “gini” for the Gini impurity. The number of features to consider when looking for the best split is also one of the classifier parameter “max_features”, its default value is “auto” that means max_features= sqrt(n_faetures). Another parameter is “max_depth” which is the maximum depth of the tree. By default, it has the value “None”. Random forest classifier also has the parameter “min_samples_split” which is the minimum number of samples required to split an internal node, it has the default value 2. “Min_weight_fraction_leaf” is one of the parameters that means the minimum weighted fraction of the sum total of weights required to be at a leaf node, it has the default value 0. Another parameter is “min_samples_leaf” which is the minimum number of samples required to be at a leaf node, it has the default value 1. One of the parameters is “max_leaf_nodes” which has the default value “None”, it means grow a tree with unlimited number of leaf nodes. “Bootstrap” is a Boolean parameter that has a default value true, that means bootstrap samples are used when building trees. Another Boolean parameter is “oob_score”, whether to use out-of-bag samples to estimate the generalization accuracy and it has the default value false. The number of jobs to run in parallel for both fit and predict can be considered as one of the parameters, it has the default value 1. Another parameter is “warm_start”, it is Boolean and its default value is false. When set to false, just fit a whole new forest, otherwise, reuse the solution of the previous call to fit and add more estimators to the ensemble. “Class_weight” is one of the parameter which means the weights associated with

the classes and it has the default value “None”. If the weights are not given, all classes are supposed to have weight one [91].

For our experiments, we had created 20 different models of random forest in two different ways.

- Create models by varying max_depth from 1 to 20.
- Create models by varying max_depth from 4 to 8 and min_samples_split from 20 to 50.

### 2.3.3 MLP Algorithm

Multi-layer perceptron ‘MLP’ classification algorithm works by the exploitation of hidden layers. Each of the nodes in the concealed stratum is a function of the nodes in the previous stratum and whatever is put out as the output is a function of the nodes in the hidden layer [85]. The perceptron usually computes a single output derived from multiple real-valued inputs by the formation of linear combinations depending on the input weights and then probably setting the output through various nonlinear activation functions.

MLP classifier has a set of parameters with different values that can be modified. The first parameter is “hidden_layer_size” which is the number of neurons in the ith hidden layer, it has the default value equal to 100. The second parameter is the activation function for the hidden layer. It can be one of the four functions “identity”, “logistic”, “tanh”, “relu”. We kept the default value “relu” the rectified linear unit function, it returns the value f(x) = max(0, x). The third parameter is the solver for weight optimization, its default value is “adam” that refers to a stochastic gradient-based optimizer. The default solver “adam” works better on relatively large datasets in terms of both validation score and training time. Another parameter is “alpha”, it has the default value 0.0001. “batch_size” is also one of the parameters it is auto by default. It represents the Size of minibatches for stochastic optimizers. If it sets to “auto”, then batch_size=min(200, n_samples).

"Learning_rate_init", the initial learning rate used, also considered as one of the parameters with the default value 0.001. It controls the step-size in updating the weights. Another parameter is the learning rate schedule to for weight updates, the default value for it is "constant" which is a constant learning rate given by 'learning_rate_init'. The maximum number of iteration is also one of the parameters of MLP classifier "max_iter" and it has the default value set to 200. However, in our experiments we changed the default value to be from 10,000 to 200,000. Another parameter is "shuffle", it is Boolean and the default value set to true, which is to shuffle samples in each iteration [91].

### 2.3.4 K-Neighbors Algorithm

K- Neighbors algorithm is a nonparametric form employed in classification and regression. In both scenarios, the input consists of K which is near the training examples in the feature space [90]. The output most of the cases depends on whether the K-NN is employed in classification or regression. KNN is known to make predictions by employing the training dataset directly. The predictions are made through the searching of the whole training set for the K in most common cases and the summarizing of the output variable for the K instances.

K-Neighbors classifier has a set of parameters with different values that can be modified. The first parameter is the number of neighbors to use, "n_neighbors" it has the default value 5. However, in our experiment we changed the number of neighbors to be from 1 to 20. Another parameter is "weights", weight function used in prediction, and the default value set to "uniform". Uniform weights means that all points in each neighborhood are weighted equally. Another parameter of K-neighbors classifier is "algorithm", which represents the algorithm that is used to compute the nearest neighbors. When it set to auto, will attempt to decide the most appropriate algorithm based on the values passed to fit method. Leaf size id also one of the parameters with the default value equal to 30, this can affect the speed of the construction and query, as well as the

memory required to store the tree. Another parameter is “metric” that represents the distance metric to use for the tree. The default metric is “minkowski”. “metric_params” is also one of the parameters which is additional keyword arguments for the metric function and the default is none. The number of parallel jobs to run for neighbors search is one of the parameters, the default is “n_jobs=1” [91].

## 2.4 Transfer Learning

We will now define a transfer learning problem, as well as our inductive transfer setting.

**Definition 2.1.1** (Transfer Learning). Given a source domain $D_S$, and learning task $T_S$, a target domain $D_T$ and learning task $T_T$, *transfer learning* aims to help improve the learning of the target predictive function $f_T(\,.\,)$ in $D_T$ using the knowledge in $D_S$ and $T_S$ where $D_S \neq D_T$ and $T_S \neq T_T$.

**Definition 2.1.2** (Inductive Transfer Learning). Given a source domain $D_S$, and learning task $T_S$, a target domain $D_T$ and learning task $T_T$, *inductive transfer learning* aims to help improve the learning of the target predictive function $f_T(\,.\,)$ in $D_T$ using the knowledge in $D_S$ and $T_S$ where $T_S \neq T_T$.

**Definition 2.1.3** (Transductive Transfer Learning). Given a source domain $D_S$, and a corresponding learning task $T_S$, a target domain $D_T$ and a corresponding learning task $T_T$, *transductive transfer learning* aims to help improve the learning of the target predictive function $f_T(\,.\,)$ in $D_T$ using the knowledge in $D_S$ and $T_S$ where $D_S \neq D_T$ and $T_S = T_T$. In addition, some unlabeled target domain data must be available at training time.

## 2.5 Related Work on Transfer Learning

Here we discuss general work on transfer learning models and methods, followed by a discussion and comparison of inductive and supervised transfer techniques.

### 2.5.1 Transfer Learning

The basic proposition of transfer learning is that it includes one target task and one or more source tasks [65]. It is not a requirement to consider labeled examples, although it is often considered. So, even the classic Semi-Supervised Learning [27, 28, 30] can be regarded as an inductive transfer learning problem in the cases where source data includes only unlabeled examples and the source domain is same as the target domain [48].

***Domain Adaptation*** (DA) is typically considered within a semi-supervised context where there is plenty of unlabeled data available [33, 48]. If we take an example of simple domain adaptation problem, a source concept could be to identify blood cell in pictures, while the target problem would be to identify the same cells on different angles. However, in DA, we have different feature and labeling spaces because of the difference between the domains.

***Multi-task learning*** (MTL) is a related learning setting whereby the goal is to produce a good hypothesis for several related learning problems simultaneously [16]. Typically, the different tasks are defined on the same input space but the probability distributions differ.

Some notable approaches for these settings are based on similarity to a common predictor [44, 34, 32, 51], finding a shared representation [26, 50, 63, 38, 45] or a shared subspace [24, 23, 25,54], as well as probabilistic approaches [52, 64, 69, 29].

The semi-supervised transfer is another interesting transfer learning setting, which has plenty of target samples in addition to labeled target samples [14, 57, 50, 35, 37, 39, 36, 40].

For a comprehensive review of these fields, the reader is referred to the works of Pan and Yang [3] and Jiang [48].

### 2.5.2 Inductive and Supervised Transfer Learning

The generic title "transfer learning" encompasses quite a few different paradigms. As noted by Levy and Markovitch [53]. The survey by Pan et al. [3] identifies the following settings, which are not mutually exclusive.

**Model Transfer:** This inductive transfer setting assumes that a good predictor for the source has been learned, resulting in an attempt to adapt the model to the target problem using a training set from the target domain. Present model transfer model methods rely on a biased regularizer [16, 67, 59, 61], on aggregating multiple source-target predictors [19, 58, 15, 62], utilizing model parameter transfer as priors [56, 65, 41], or by feature weight estimation [42, 60].

**Instance Transfer**: Working on a supervised inductive transfer learning problem, one assumes certain instances of the source data can be used as examples in the target domain. Under this assumption, it is better to take some of the source data "as is", and the problem reduces to identifying the relevant instances and ignoring the irrelevant ones, using a process of elimination or weighting. Boosting-based instance weighing is common practice in this category [31, 18, 68, 17], as is instance elimination (and sub-sampling) [49, 22], but other techniques exist for utilizing the source information in different ways [67, 32, 63, 46].

**Feature Transfer**: Algorithm working in this supervised transfer learning setting attempts to learn a feature mapping or weighing if we assume that there exists some partial relation between the source and target features. These techniques represent an attempt to find the "common denominators" of the learning tasks, matching features, or combinations of features, to identify meaning in partial information. Norm optimization [57, 43, 47], manipulating and combining features [41, 55, 53] are some of the standard techniques to address this problem.

### 2.6 Evaluation Metrics

In this thesis, we train each model with training data source and each trained model is tested using the test data source. Here we have used accuracy, precision, and recall for evaluating and comparing the performance of the trained model with test data source. In our experiments, we have used the 5-fold cross validations for evaluating trained models so we have used the mean value of accuracy, precision, and recall to record the performance of each model.

#### 2.6.1 Accuracy

Accuracy is commonly defined as the statistical measure of the difference between a result and an actual value.

if $\hat{y}_i$ is the predicted output of the $i_{th}$ sample and $y_i$ is the corresponding actual value, then the accuracy over $n_{sample}$ is defined as:

$$accuracy(\mathrm{y}, \hat{y}) = \frac{1}{n_{sample}} \sum_{i=0}^{n_{sample}-1} 1(y_i = \hat{y}_i)$$

#### 2.6.2 Precision and Recall

Before the understanding precision and recall we need to know the meaning of true positive, true negative, false positive and false negative. In general, positive is equivalent to identified and negative is equivalent to rejected. So

- True positive = correctly identified
- False positive = incorrectly identified
- True negative = correctly rejected
- False negative = incorrectly rejected.

For example, while predicting a patient of having heart disease or not it can have following output:

- True positive: heart disease patient identified as heart disease patient
- False positive: non-heart disease patient incorrectly identified as heart disease patient
- True negative: non-heart disease patient identified as non-heart disease patient
- False negative: heart disease patient identified as non-heart disease patient

Precision is a measure of the relevancy of the result whereas recall is a measure of how many truly relevant results are returned.

Precision P is defined as the number of true positives ($T_p$) over the number of true positives and the number of false positives ($F_p$) combined.

$$P=\frac{Tp}{Tp+Fp}$$

Recall R is defined as the number of true positives ($T_p$) over the number of true positives and the number of false negatives ($F_n$) combined.

$$R=\frac{Tp}{Tp+Fn}$$

We have used the 'average' parameter in the calculations. It is required for multiclass/multi label targets. Below are the types of averaging that we performed.

**'micro'**: Calculate metrics globally by counting the total true positives, false negatives and false positives.

**'macro'**: This does not take label imbalance into account. Calculate metrics for each label, and also finds their unweighted mean.

**'weighted'**: This changes the 'macro' to account for label imbalance. Calculate metrics for each label, and find their average, weighted by the support.

### 2.6.3 Visual Representation of Evaluation

#### 2.6.3.1 Line Graph

A line graph is very useful in cases where you want to see the rate of change clearly between individual data points. We have plotted line graphs with the model number on the x-axis. For each algorithm, we have plotted graphs with accuracy, precision or recall on the y-axis.

#### 2.6.3.2 Box Chart

Box plot depicts a group of numerical data using quartiles and the lines extending vertically from boxes indicate the variability. We have plotted the graph depicting algorithm on x-axis and accuracy/precision/recall on the y-axis.

#### 2.6.3.3 Roc Chart

Receiver operating characteristic (ROC) curve is created by plotting the true positive rate (also known as sensitivity or recall) vs false positive rate (1-specificity).

$$\text{TPR}=\frac{TP}{TP+FN}$$

And

$$\text{FPR}=\frac{FP}{FP+TN}$$

The area under the ROC curve is a measure of how well a parameter can differentiate the two diagnostic groups.

CHAPTER III

METHODOLOGY

In this chapter, we have mention detail architecture and algorithm followed for performing the experiments for this thesis. This architectures and algorithms were followed for all the models and datasets during the experiments.

### 3.1 Machine Learning Architecture

This architecture includes the traditional procedure followed during machine learning. In this process, we consider all the feature of the dataset for training and testing the model.

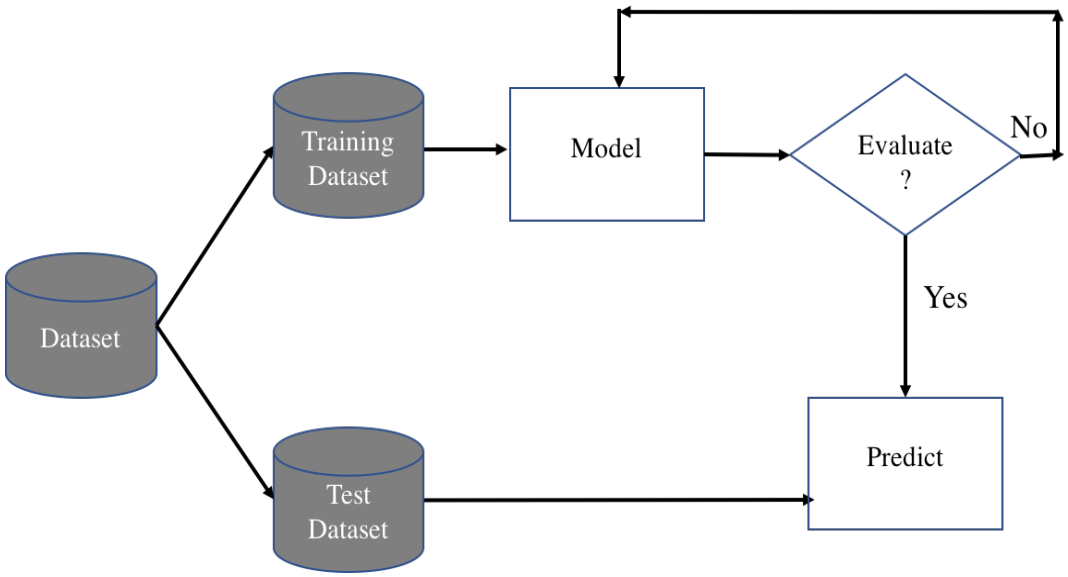


Figure 3.1. Supervised Machine Learning Architecture

To implement the above architecture, we followed the following steps:

- Pre-processing the data based on following criteria:

        - If the feature values contains string or alphanumerical characters, perform label encoding on the feature data and store label encoder based on the feature name
        - If the feature values are numerical copy as it is
    - Split data randomly into 80% for training dataset and 20% for test dataset.
    - Create 20 different models of Decision Tree, Random Forest, KNN and MLP based on max_depth [ 1 to 20 ], n_neighbor [1 to 20] and max_iteration [ 10,000 to 200,000]
    - Repeat following steps for 20 different models of Decision Tree/Random Forest/KNN/MLP Algorithms.
        - Train the model with train dataset
        - Predict the evaluation criteria (accuracy, precision, recall) for each model using test data set.
    - Find the best model base on evaluation criteria for each algorithm (DT/RF/KNN/MLP)

**3.2 Transfer Learning Architecture**

In this thesis, we introduce a simple algorithm where we applied feature transfer learning. Firstly, we identify the feature importance based on the source dataset. Secondly, we use these identified features for training model on target dataset and perform a prediction on target test dataset.

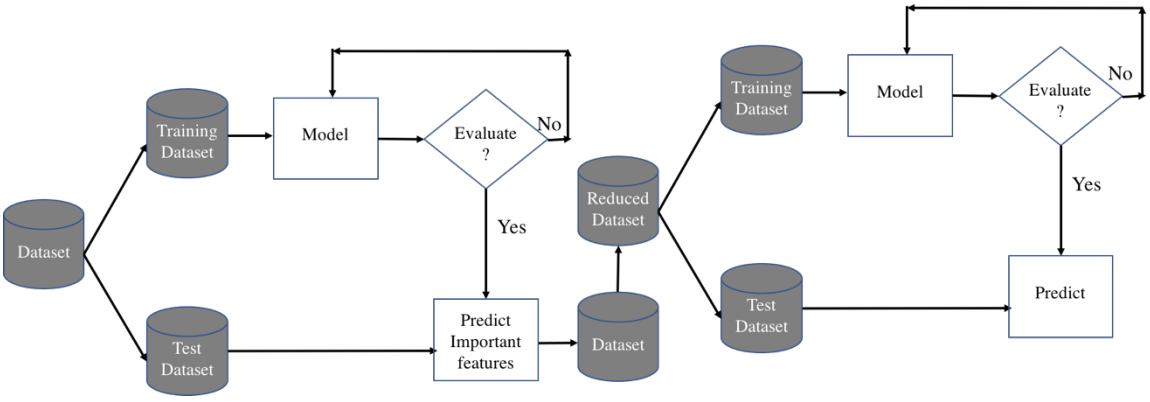


Figure 3.2. Transfer Learning Architecture

To implement the above architecture, we followed the following steps:

- Pre-processing the data based on following criteria:
    - If the feature values contains string or alphanumerical characters, perform label encoding on the feature data and store label encoder based on the feature name
    - If the feature value is numerical copy as it is
- Split data randomly into 80% for training dataset and 20%test data set.
- Create 20 different models of Decision Tree and Random Forest based on max_depth changed from 1 to 20.
- Repeat following steps for 20 different models of Decision Tree algorithm.
    - Train the model with train dataset
    - Predict the accuracy of model with test dataset
    - Store the feature importance of the trained model in ascending order [ most important first]
- Find top 10 important features of the Decision Tree algorithm.
    - Choose top 10 important features for each model.
    - Calculate the frequency count for each feature from 20 different models.

    - Choose the top 10 features with highest frequency count.
- Perform the transfer learning by selecting only top 10 important features data [identified in early step] from the pre-processed data.
- Split data randomly into 80% for training dataset and 20%test data set.
- Create 20 different models of Decision Tree, Random Forest, KNN and MLP based on max_depth [ 1 to 20 ], n_neighbor [1 to 20] and max_iteration [ 10,000 to 200,000]
- Repeat following steps for 20 different models of Decision Tree/Random Forest/KNN/MLP Algorithms.
    - Train the model with train dataset
    - Predict the evaluation criteria (accuracy, precision, recall) for each model using test data set.
- Find the best model base on evaluation criteria for each algorithm (DT/RF/KNN/MLP)

**3.3 Suggested Feature Architecture**

In this thesis, we also got suggested important features from an expert. Asper expert, these suggested important features are needed for the identification of the probability of having heart disease or probability of getting readmission. So, using that suggested feature, we performed the experiment by using following architecture.

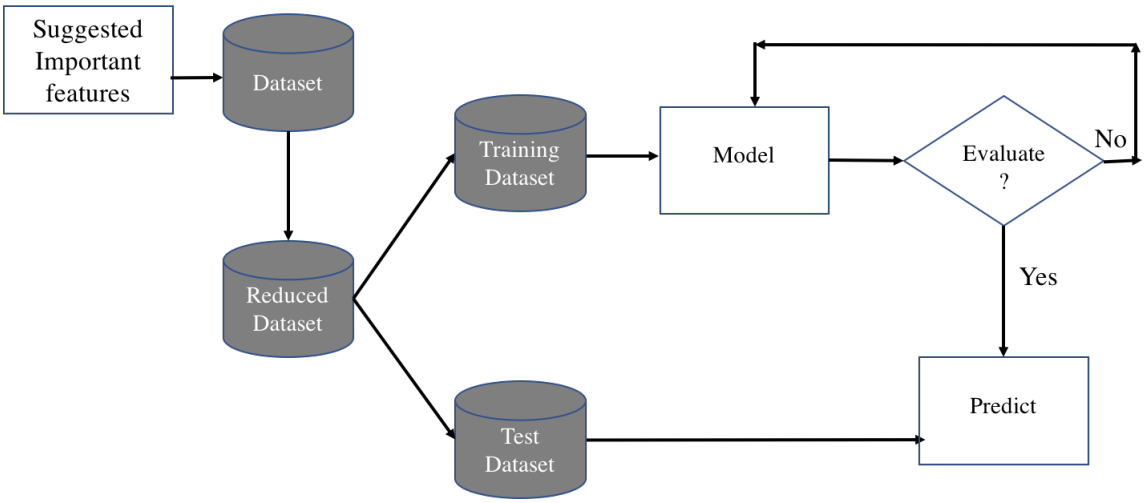


Figure 3.3. Suggested Feature Machine Learning Architecture

To implement the above architecture, we followed the following steps:

- Create a data, by selecting only suggested features data from the data source.
- Pre-processing the data based on following criteria:
    - If the feature values contains string or alphanumerical characters, perform label encoding on the feature data and store label encoder based on the feature name
    - If the feature values are numerical, copy as it is
- Split data randomly into 80% for training dataset and 20% for test dataset.
- Create 20 different models of Decision Tree, Random Forest, KNN and MLP based on max_depth [1 to 20], n_neighbor [1 to 20] and max_iteration [10,000 to 200,000]
- Repeat following steps for 20 different models of Decision Tree/Random Forest/KNN/MLP Algorithms.
    - Train the model with train dataset
    - Predict the evaluation criteria (accuracy, precision, recall) for each model using test data set.

- Find the best model base on evaluation criteria for each algorithm (DT/RF/KNN/MLP)

The suggested feature for dataset are as follow:

For Heart disease dataset suggested features are: [ "age_1", "max_glu_serum_1", "A1Cresult_1", "diabetesMed_1", "gender_1", "time_in_hospital_1","num_medications_1", "readmitted_1", "race_1", "Kidney Problem", "Ulcers, Toe, Foot, Leg æAmputation" ]

For Readmission dataset suggested features are: [ "age", "max_glu_serum", "A1Cresult", "diabetesMed", "gender", "time_in_hospital", "num_medications", "race" ]

### 3.4 Transfer Learning Combined With Suggested Feature Architecture

Here we combined the important features identified during transfer learning with the suggested features from the expert and performed the experiments on the dataset. We used the following architecture to perform the experiments.

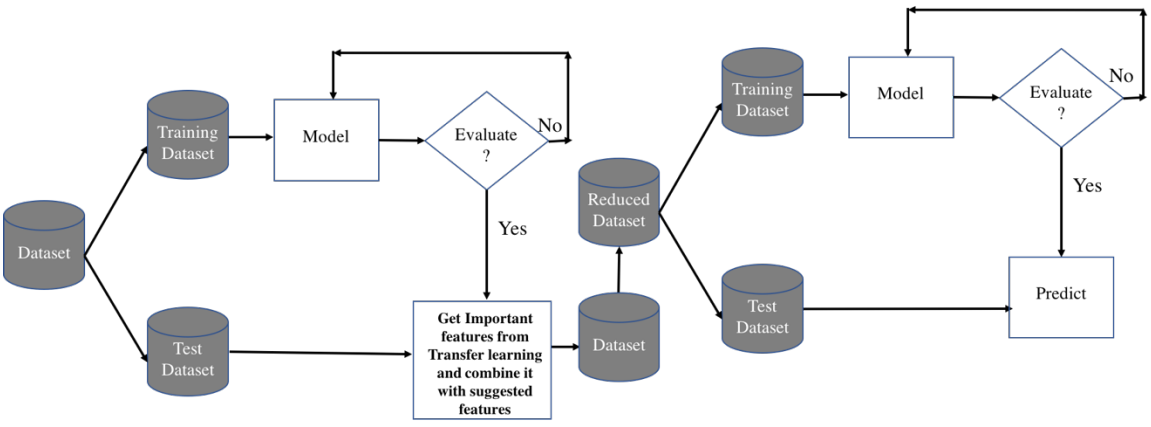


Figure 3.4. Transfer Learning Combined with Suggested Features Architecture

Implemented above architecture, using following steps:

- Pre-processing the data based on following criteria:
    - If the feature values contains string or alphanumerical characters, perform label encoding on the feature data and store label encoder based on the feature name

    - If the feature value is numerical copy as it is
- Split data randomly into 80% for training dataset and 20%test data set.
- Create 20 different models of Decision Tree and Random Forest based on max_depth changed from 1 to 20.
- Repeat following steps for 20 different models of Decision Tree algorithm.
    - Train the model with train dataset
    - Predict the accuracy of model with test dataset
    - Store the feature importance of the trained model in ascending order [most important first]
- Find top 10 important features of the Decision Tree algorithm.
    - Choose top 10 important features for each model.
    - Calculate the frequency count for each feature from 20 different models.
    - Choose the top 10 features with highest frequency count.
- *Create important features list by combining the top 10 important features with suggested important features.*
- Perform the transfer learning by selecting only important features list from the pre-processed data.
- Split data randomly into 80% for training dataset and 20%test data set.
- Create 20 different models of Decision Tree, Random Forest, KNN and MLP based on max_depth [ 1 to 20 ], n_neighbor [1 to 20] and max_iteration [ 10,000 to 200,000]
- Repeat following steps for 20 different models of Decision Tree/Random Forest/KNN/MLP Algorithms.

        - Train the model with train dataset
        - Predict the evaluation criteria (accuracy, precision, recall) for each model using test data set.
- Find the best model base on evaluation criteria for each algorithm (DT/RF/KNN/MLP)

# CHAPTER IV

# RESULTS AND FINDINGS

In this work, the machine learning algorithms that we used showed significant results by using several datasets. These results have been compared to observe which algorithm performed better in terms of the evaluation criteria [section 2.6]. Also, we found the best model for each algorithm. In this chapter, we also showed that the transfer learning technique to train the models and then we compared the models with the evaluation criteria. Also, we compared the models with transfer learning technique to models without transfer learning for each dataset.

The dataset that we used in this study is available as a Supplementary Material available online at http;//dx.doi.org/10.1155/2014/781670. In presenting the dataset, we used the structure that is used in the published paper ("Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records", 2014).

## 4.1 Result

As a result of experiments, we compared the evaluation criteria of each model that either used or did not use transferring learning technique, which we used for different data sources. For these experiments, we had a heart disease dataset and readmission dataset.

For each data set, first of all we created different models of each algorithm by training models that use all the features available with the dataset. Then, we found the best model for each algorithm.

Second of all, we created models for each algorithm using transfer learning technique. Each model was trained using only important features identified during the transfer learning technique. Then, we again found the best model for each algorithm.

We also used the suggested important features identified by the expert to train and test the models and we also compared each model using that. Again, new models were created by training the models with the features suggested by expert combined with the important features identified during the transfer learning technique. For both techniques, best model of each algorithms was identified by comparing with evaluation criteria.

Finally, we compare all the best models of each technique. This comparison also showed whether efficiency of the model is impacted by considering only features identified during transfer learning or by suggested features.

### 4.2 Heart Disease Dataset

We did different experiments on the heart disease dataset by considering the machine learning algorithms and transfer learning technique. The description of the heart disease dataset is explained at appendix A.

### 4.3 Results and Finding in Heart Disease Dataset

In this section, we mention detail results obtained for heart disease dataset by following methodologies specified in chapter 3.

#### 4.3.1 Using All the Features of the Dataset

In this technique, we created the 20 different models of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithm by modifying the coefficient of the model. Here, we trained each model using all the features available in the dataset, and then we estimated each trained model performance by calculating evaluation metrics using grid search with 5-fold cross validation. We used the different techniques to compare the different models of the different algorithms as shown below.

#### 4.3.1.1 Line Graph

Using Grid Search with 5-fold cross validation, we have calculated the evaluation metrics for all the models of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithm. Following line diagrams shows value of evaluation metrics based on varying coefficient of each algorithm.

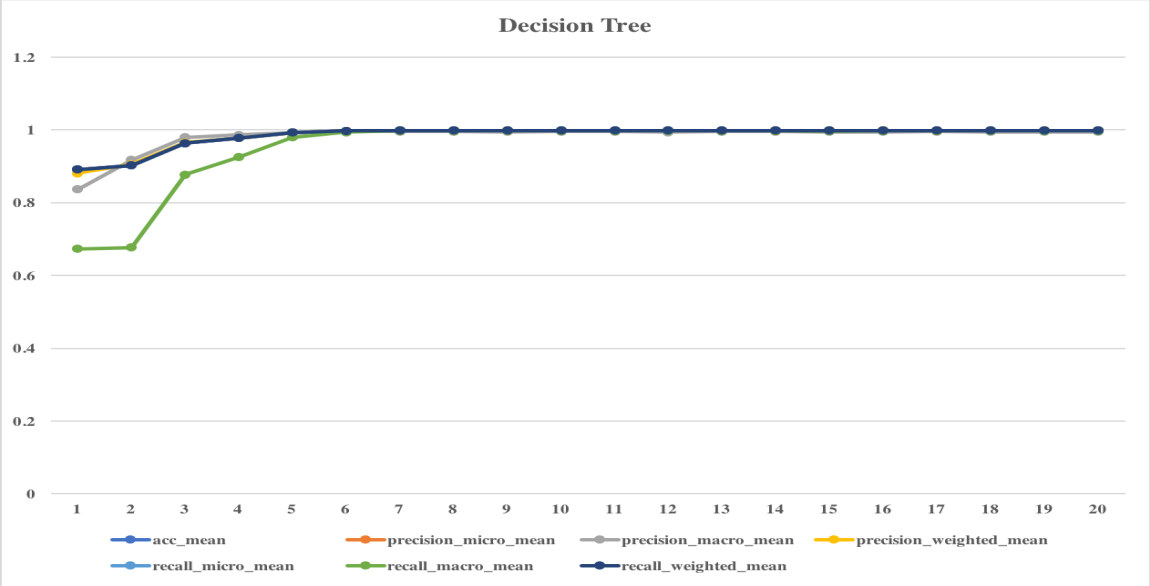


Figure 4.1. Line Graph of Decision Tree with Varying Max-Depth for Heart Disease Dataset Using All Features

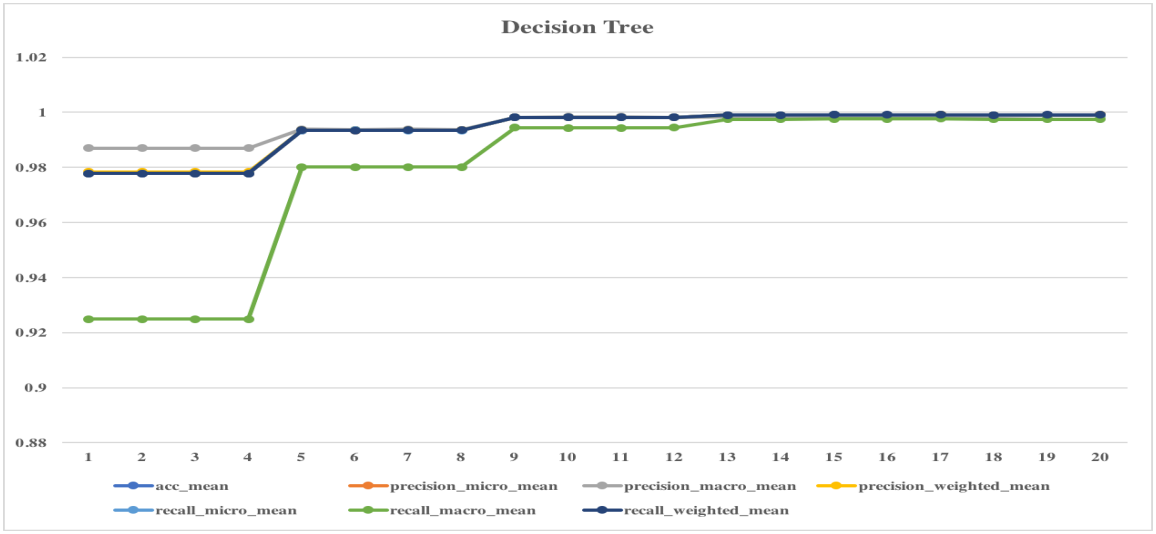


Figure 4.2. Line Graph of Decision Tree with Varying Max_Depth and Min_Sample_Split for Heart Disease Dataset Using All Features

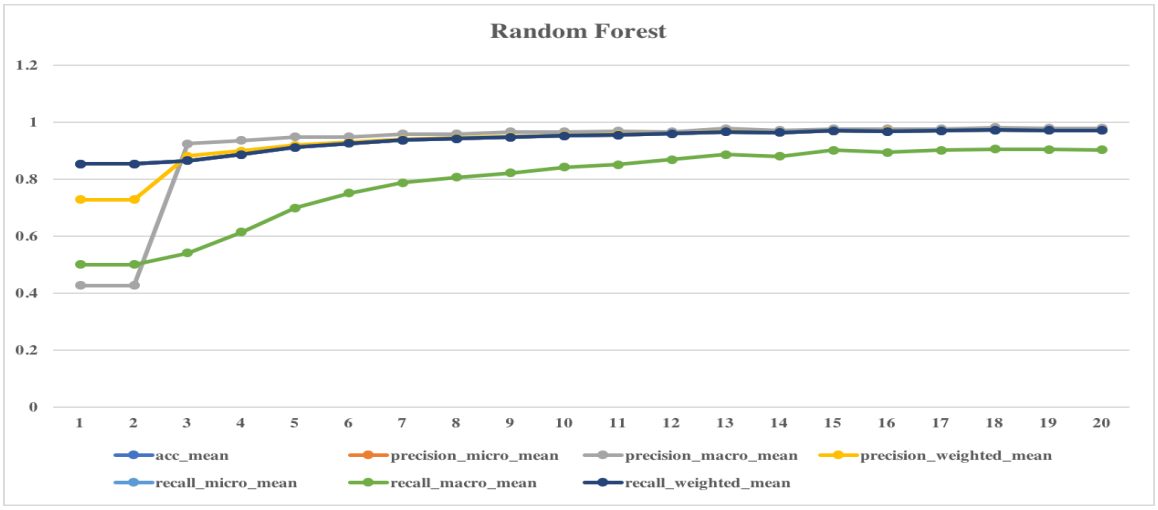


Figure 4.3. Line Graph of Random Forest with Varying Max-Depth for Heart Disease Dataset Using All Features

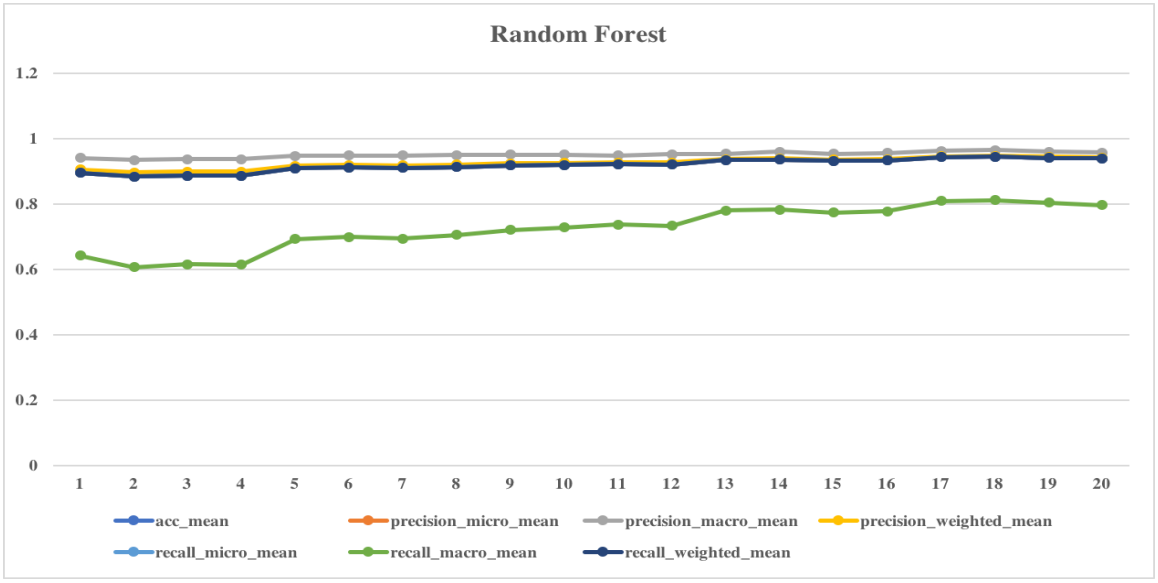


Figure 4.4. Line Graph of Random Forest with Varying Max_Depth and Min_Sample_Split for Heart Disease Dataset Using All Features

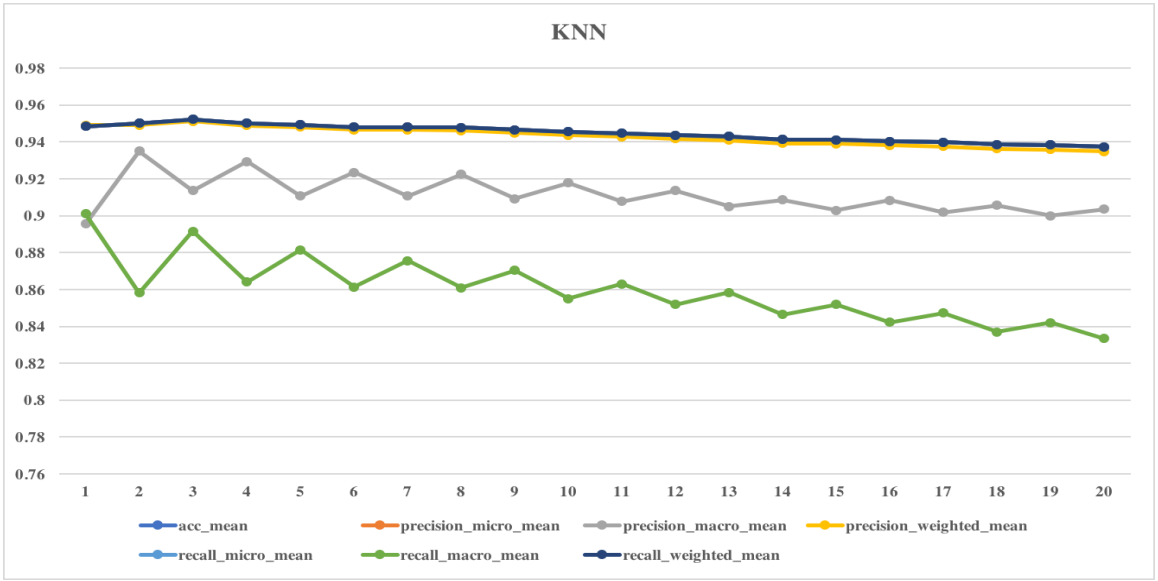


Figure 4.5. Line Graph of KNN with Varying N_Neighbor for Heart Disease Dataset Using All Features

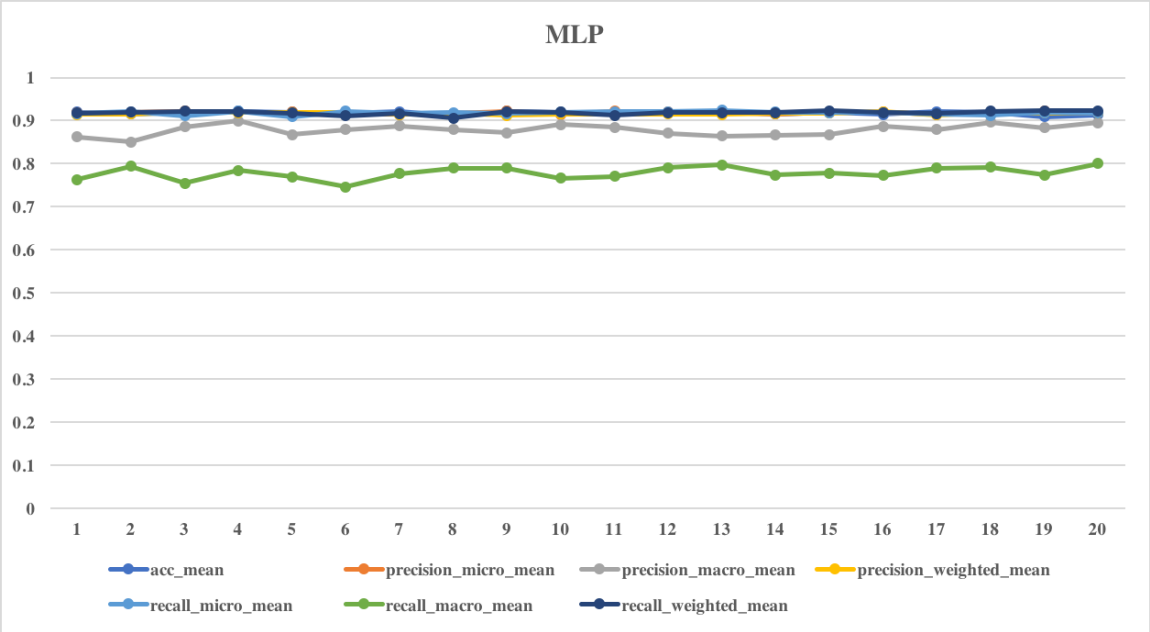


Figure 4.6. Line Graph of MLP with Varying Max_Iteration for Heart Disease Dataset Using All Features

#### 4.3.1.2 Box Plot

Following box plot shows the comparison of the evaluation metrics values for all the models created by varying max_depth of Decision Tree, max_depth of Random Forest, n_neighbor of K-Nearest Neighbor and max_iteration of MLP algorithms for heart disease dataset using all features.

Table 4.1. Accuracy Value for Heart Disease Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.891079812 | 0.852999478 | 0.937350026 | 0.909389671 |
| First Quartile (Q1) | 0.996557121 | 0.921622327 | 0.940923318 | 0.915219092 |
| Median Value | 0.998043818 | 0.953390715 | 0.94504434 | 0.919353156 |
| Third Quartile(Q3) | 0.998135107 | 0.967749087 | 0.948226395 | 0.920174752 |
| Max Value | 0.998435055 | 0.971465832 | 0.952164841 | 0.922222222 |
| Box 1-hidden (Q1) | 0.996557121 | 0.921622327 | 0.940923318 | 0.915219092 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Box 2 (Median - Q1) | 0.001486698 | 0.031768388 | 0.004121022 | 0.004134064 |
| Box 3 (Q3- Median) | 9.12885E-05 | 0.014358372 | 0.003182055 | 0.000821596 |
| Whisker Top (Max- Q3) | 0.000299948 | 0.003716745 | 0.003938445 | 0.00204747 |
| Whisker Bottom (Q1- Min) | 0.105477308 | 0.068622848 | 0.003573292 | 0.005829421 |

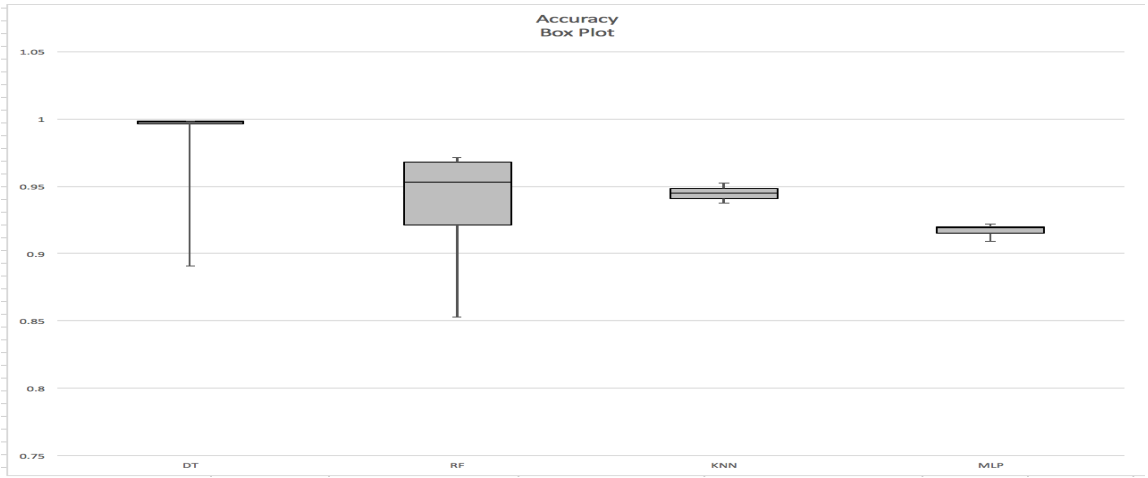


Figure 4.7. Accuracy Box Plot for Heart Disease Dataset Using All Features

Table 4.2. Precision Macro Value for Heart Disease Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.837102924 | 0.426499739 | 0.895611149 | 0.850541438 |
| First Quartile (Q1) | 0.994177712 | 0.948092673 | 0.904552353 | 0.867694403 |
| Median Value | 0.994901476 | 0.966035886 | 0.908844417 | 0.87942315 |
| Third Quartile(Q3) | 0.995275639 | 0.976302151 | 0.914618452 | 0.886819007 |
| Max Value | 0.996739538 | 0.980816173 | 0.935128792 | 0.899345017 |
| Box 1-hidden (Q1) | 0.994177712 | 0.948092673 | 0.904552353 | 0.867694403 |
| Box 2 (Median - Q1) | 0.000723764 | 0.017943212 | 0.004292064 | 0.011728747 |
| Box 3 (Q3- Median) | 0.000374162 | 0.010266265 | 0.005774035 | 0.007395857 |
| Whisker Top (Max- Q3) | 0.0014639 | 0.004514022 | 0.02051034 | 0.01252601 |
| Whisker Bottom (Q1- Min) | 0.157074788 | 0.521592934 | 0.008941204 | 0.017152965 |

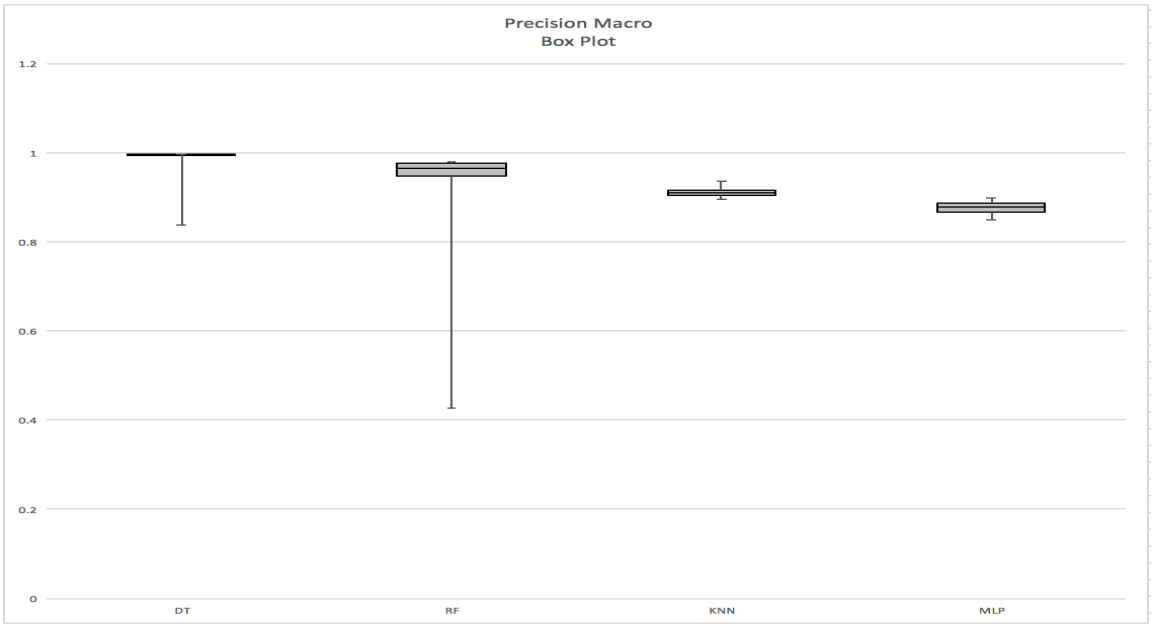


Figure 4.8. Precision Macro Box Plot for Heart Disease Dataset Using All Features

Table 4.3. Precision Micro Value for Heart Disease Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.891079812 | 0.852999478 | 0.937350026 | 0.914136672 |
| First Quartile (Q1) | 0.996557121 | 0.921622327 | 0.940923318 | 0.915832029 |
| Median Value | 0.997965571 | 0.953390715 | 0.94504434 | 0.918492436 |
| Third Quartile(Q3) | 0.998069901 | 0.967749087 | 0.948226395 | 0.91982264 |
| Max Value | 0.99838289 | 0.971465832 | 0.952164841 | 0.921804903 |
| Box 1-hidden (Q1) | 0.996557121 | 0.921622327 | 0.940923318 | 0.915832029 |
| Box 2 (Median - Q1) | 0.001408451 | 0.031768388 | 0.004121022 | 0.002660407 |
| Box 3 (Q3- Median) | 0.00010433 | 0.014358372 | 0.003182055 | 0.001330203 |
| Whisker Top (Max- Q3) | 0.000312989 | 0.003716745 | 0.003938445 | 0.001982264 |
| Whisker Bottom (Q1- Min) | 0.105477308 | 0.068622848 | 0.003573292 | 0.001695357 |

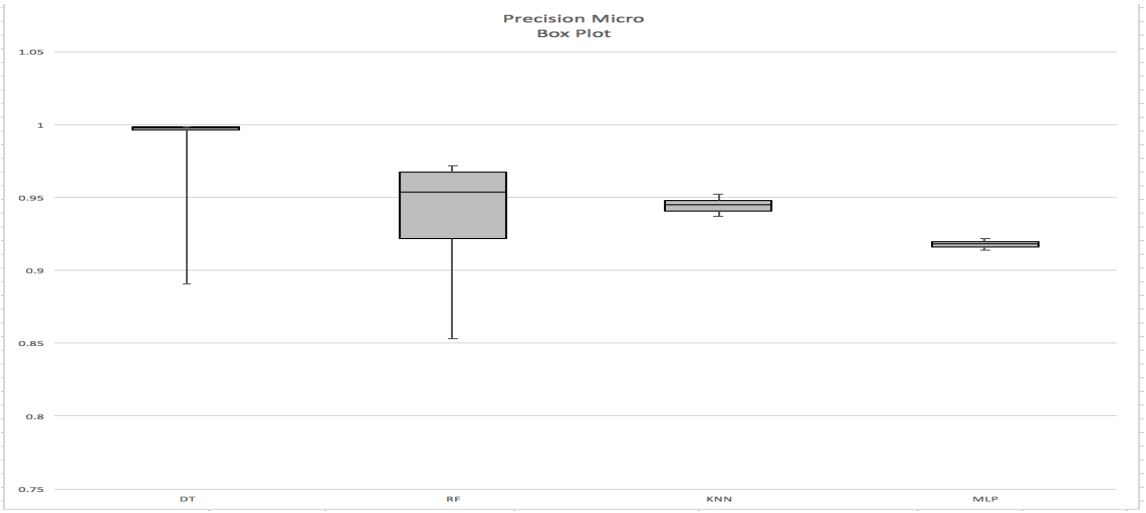


Figure 4.9. Precision Micro Box Plot for Heart Disease Dataset Using All Features

Table 4.4 Precision Weighted Value for Heart Disease Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.881309714 | 0.72760812 | 0.934777559 | 0.912218646 |
| First Quartile (Q1) | 0.996652962 | 0.926388327 | 0.93874975 | 0.914911471 |
| Median Value | 0.997952558 | 0.954978744 | 0.943239351 | 0.916110954 |
| Third Quartile(Q3) | 0.998108034 | 0.968409705 | 0.946982708 | 0.918394008 |
| Max Value | 0.998287292 | 0.972169671 | 0.951244556 | 0.920852791 |
| Box 1-hidden (Q1) | 0.996652962 | 0.926388327 | 0.93874975 | 0.914911471 |
| Box 2 (Median - Q1) | 0.001299596 | 0.028590417 | 0.004489601 | 0.001199483 |
| Box 3 (Q3- Median) | 0.000155476 | 0.013430962 | 0.003743357 | 0.002283054 |
| Whisker Top (Max- Q3) | 0.000179258 | 0.003759966 | 0.004261848 | 0.002458783 |
| Whisker Bottom (Q1- Min) | 0.115343248 | 0.198780207 | 0.003972191 | 0.002692824 |

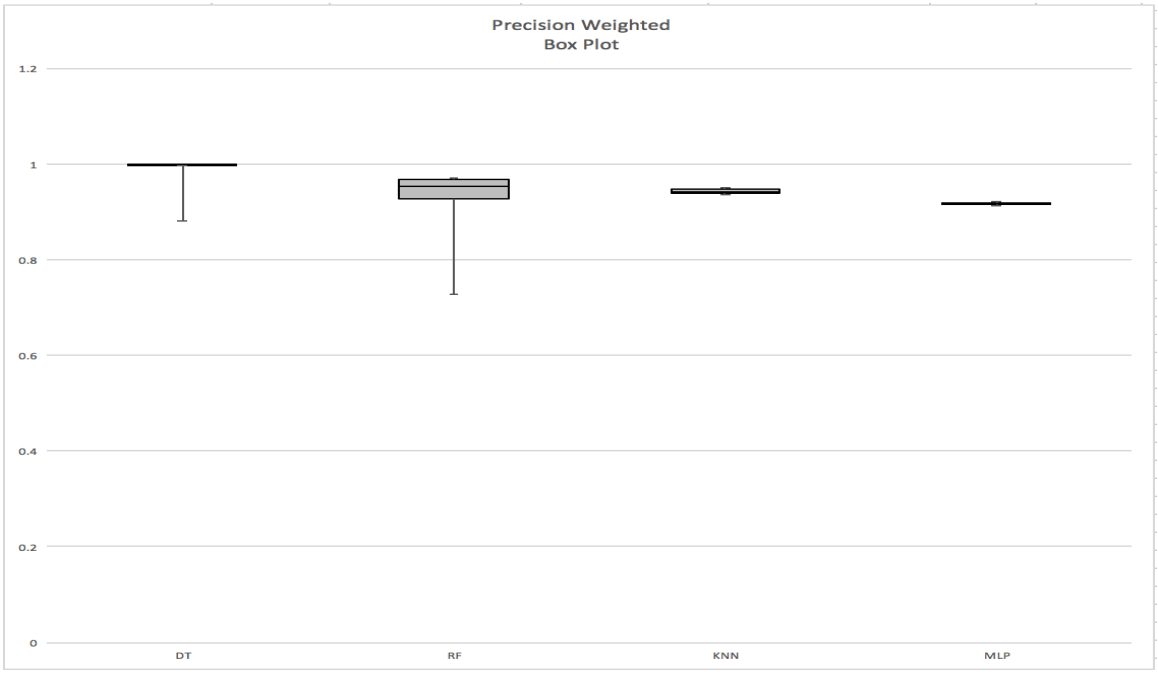


Figure 4.10. Precision Weighted Box Plot for Heart Disease Dataset Using All Features

Table 4.5. Recall Macro Value for Heart Disease Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.673276336 | 0.5 | 0.833447012 | 0.745708297 |
| First Quartile (Q1) | 0.99066504 | 0.737312514 | 0.847023712 | 0.77004462 |
| Median Value | 0.997060532 | 0.845712335 | 0.858288954 | 0.777728363 |
| Third Quartile(Q3) | 0.997260795 | 0.895066348 | 0.865620745 | 0.789971378 |
| Max Value | 0.997467173 | 0.905284003 | 0.9009091 | 0.800657107 |
| Box 1-hidden (Q1) | 0.99066504 | 0.737312514 | 0.847023712 | 0.77004462 |
| Box 2 (Median - Q1) | 0.006395492 | 0.108399821 | 0.011265241 | 0.007683743 |
| Box 3 (Q3- Median) | 0.000200263 | 0.049354013 | 0.007331791 | 0.012243015 |
| Whisker Top (Max- Q3) | 0.000206378 | 0.010217656 | 0.035288355 | 0.010685729 |
| Whisker Bottom (Q1- Min) | 0.317388705 | 0.237312514 | 0.0135767 | 0.024336323 |

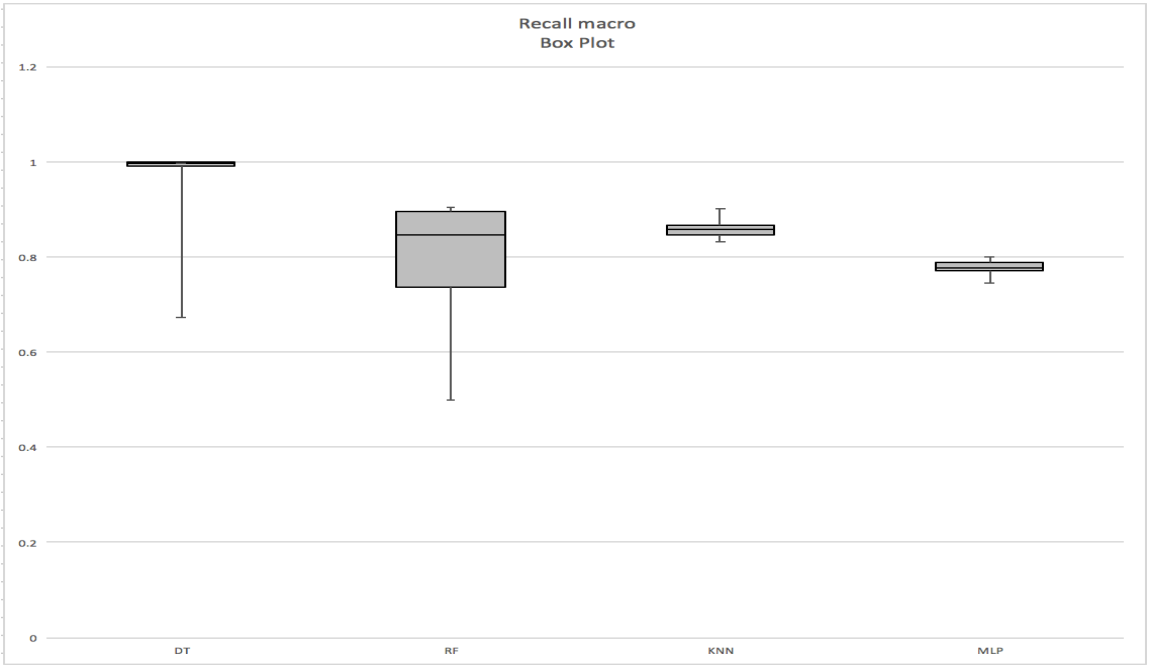


Figure 4.11. Recall Macro Box Plot for Heart Disease Dataset Using All Features

Table 4.6. Recall Micro Value for Heart Disease Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.891079812 | 0.852999478 | 0.937350026 | 0.909024517 |
| First Quartile (Q1) | 0.996517997 | 0.921622327 | 0.940923318 | 0.916118936 |
| Median Value | 0.997939489 | 0.953390715 | 0.94504434 | 0.918622848 |
| Third Quartile(Q3) | 0.998148148 | 0.967749087 | 0.948226395 | 0.920187793 |
| Max Value | 0.998435055 | 0.971465832 | 0.952164841 | 0.923682838 |
| Box 1-hidden (Q1) | 0.996517997 | 0.921622327 | 0.940923318 | 0.916118936 |
| Box 2 (Median - Q1) | 0.001421492 | 0.031768388 | 0.004121022 | 0.002503912 |
| Box 3 (Q3- Median) | 0.000208659 | 0.014358372 | 0.003182055 | 0.001564945 |
| Whisker Top (Max- Q3) | 0.000286907 | 0.003716745 | 0.003938445 | 0.003495044 |
| Whisker Bottom (Q1- Min) | 0.105438185 | 0.068622848 | 0.003573292 | 0.007094418 |

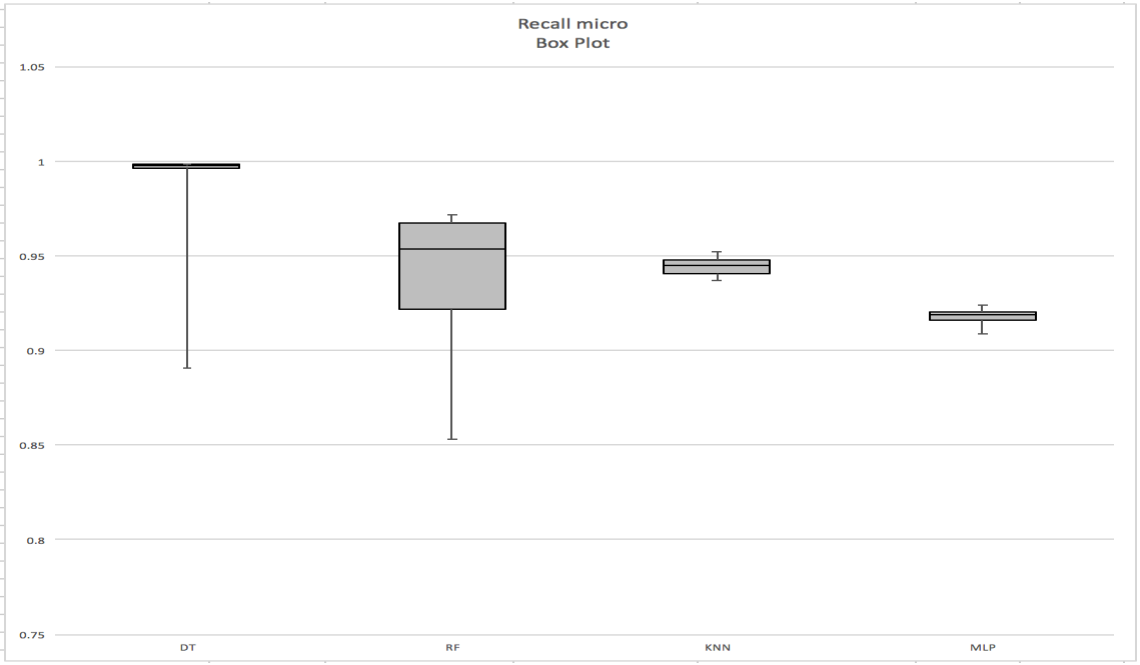


Figure 4.12. Recall Micro Box Plot for Heart Disease Dataset Using All Features

Table 4.7. Recall Weighted Value for Heart Disease Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.891079812 | 0.852999478 | 0.937350026 | 0.905946792 |
| First Quartile (Q1) | 0.996609285 | 0.921622327 | 0.940923318 | 0.917423057 |
| Median Value | 0.997939489 | 0.953390715 | 0.94504434 | 0.91896192 |
| Third Quartile(Q3) | 0.998069901 | 0.967749087 | 0.948226395 | 0.921309338 |
| Max Value | 0.998330725 | 0.971465832 | 0.952164841 | 0.923109025 |
| Box 1-hidden (Q1) | 0.996609285 | 0.921622327 | 0.940923318 | 0.917423057 |
| Box 2 (Median - Q1) | 0.001330203 | 0.031768388 | 0.004121022 | 0.001538863 |
| Box 3 (Q3- Median) | 0.000130412 | 0.014358372 | 0.003182055 | 0.002347418 |
| Whisker Top (Max- Q3) | 0.000260824 | 0.003716745 | 0.003938445 | 0.001799687 |
| Whisker Bottom (Q1- Min) | 0.105529473 | 0.068622848 | 0.003573292 | 0.011476265 |

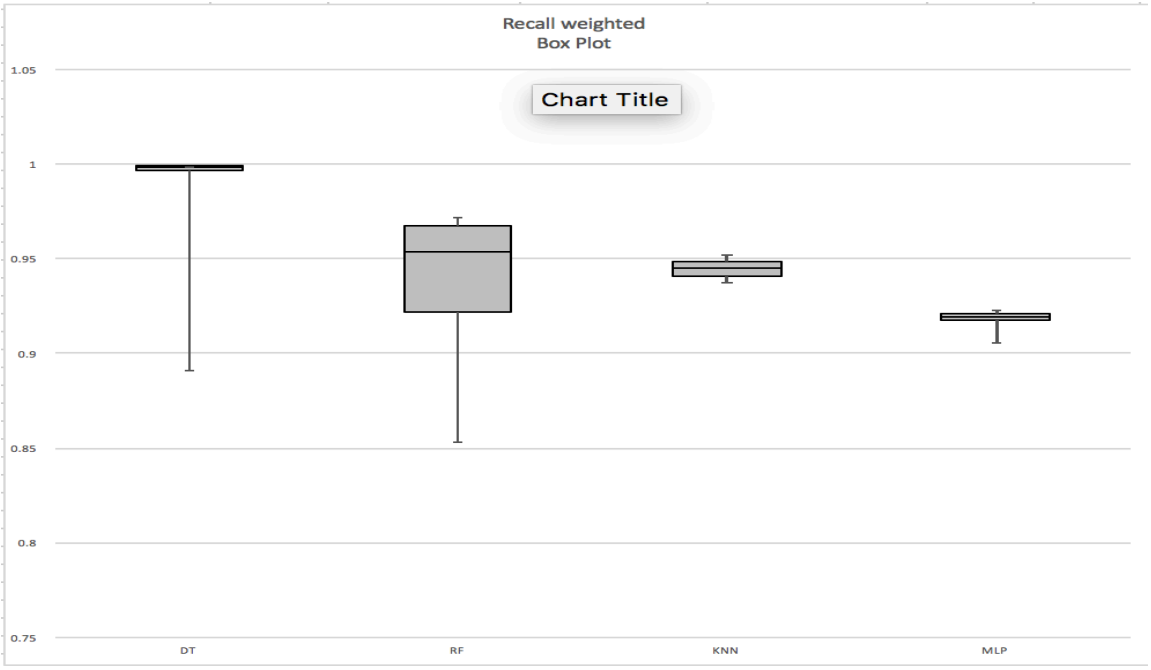


Figure 4.13. Recall Weighted Box Plot for Heart Disease Dataset Using All Features

#### 4.3.1.3 Best Model

The following diagram shows the best model of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithm for each evaluation metrics.

| Algorithm | Accuracy | | Precision Micro | | Precision Macro | | Precision Weighted | | Recall Micro | | Recall Macro | | Recall Weighted | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value |
| Decission Tree | max_depth: 7 | 0.998435055 | max_depth: 7 | 0.99838289 | max_depth: 6 | 0.996739538 | max_depth: 15 | 0.998287292 | max_depth: 9 | 0.998435055 | max_depth: 16 | 0.997467173 | max_depth: 8 | 0.998330725 |
| Random Forest | max_depth: 18 | 0.971465832 | max_depth: 18 | 0.971465832 | max_depth: 18 | 0.980816173 | max_depth: 18 | 0.972169671 | max_depth: 18 | 0.971465832 | max_depth: 18 | 0.905284003 | max_depth: 18 | 0.971465832 |
| KNN | n_neighbors: 3 | 0.952164841 | n_neighbors: 3 | 0.952164841 | n_neighbors: 2 | 0.935128792 | n_neighbors: 3 | 0.951244556 | n_neighbors: 3 | 0.952164841 | n_neighbors: 1 | 0.9009091 | n_neighbors: 3 | 0.952164841 |
| MLP | max_iter: 90000 | 0.922222222 | max_iter: 110000 | 0.921804903 | max_iter: 40000 | 0.899345017 | max_iter: 160000 | 0.920852791 | max_iter: 130000 | 0.923682838 | max_iter: 200000 | 0.800657107 | max_iter: 190000 | 0.923109025 |

Figure 4.14. Best Models for Heart Disease Dataset Using All Features

#### 4.3.1.4 ROC

Here we compare the ROC curve of the best models of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithm.

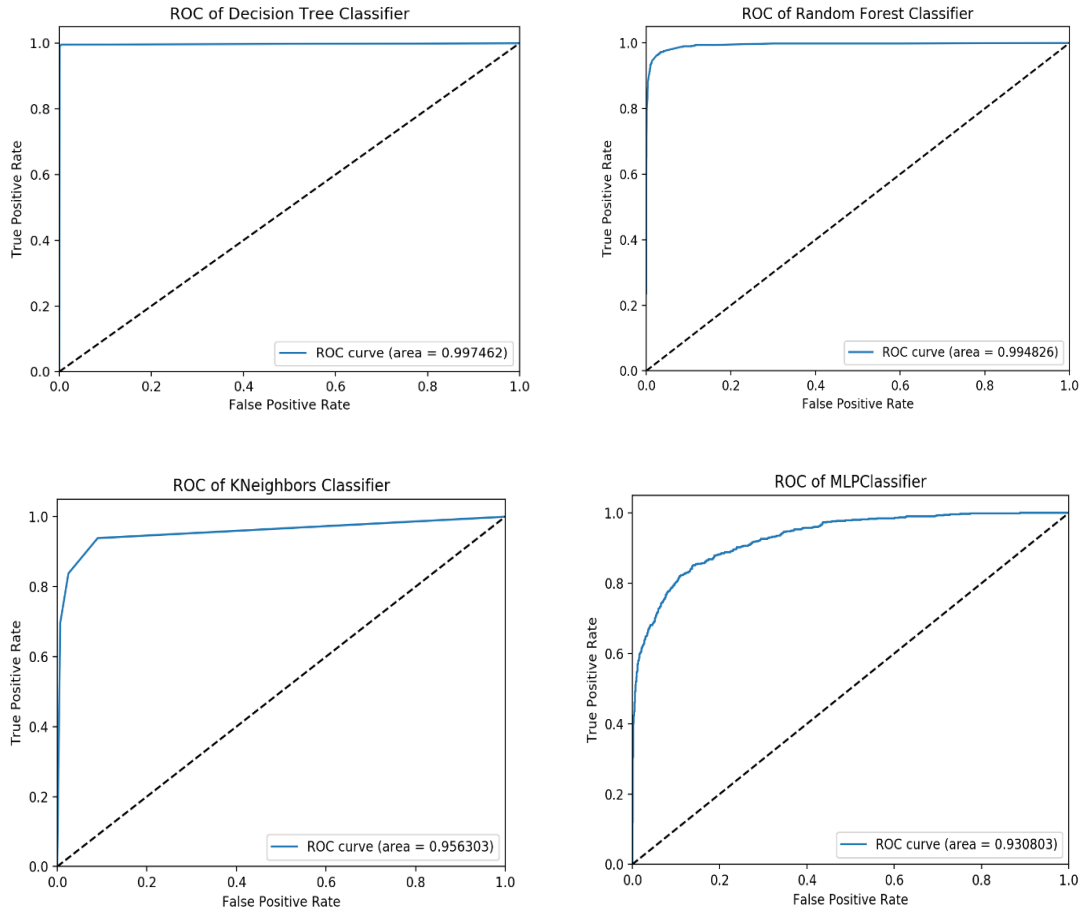


Figure 4.15. ROC Curve of Best Models for Heart Disease Dataset Using All Features

### 4.3.2 Using Transfer Learning

In transfer learning technique, for heart disease dataset we identified the top 10 important features during transfer learning using decision tree. And these top 10 features were only used for training all the models of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithm to demonstrate the transfer learning in this experiment. For estimating the evaluation metrics, we also used the grid search with 5-fold cross validation and compared the models.

#### 4.3.2.1 Line Graph

The following line graph is used to show the varying evaluation metrics value for all models of each algorithm.

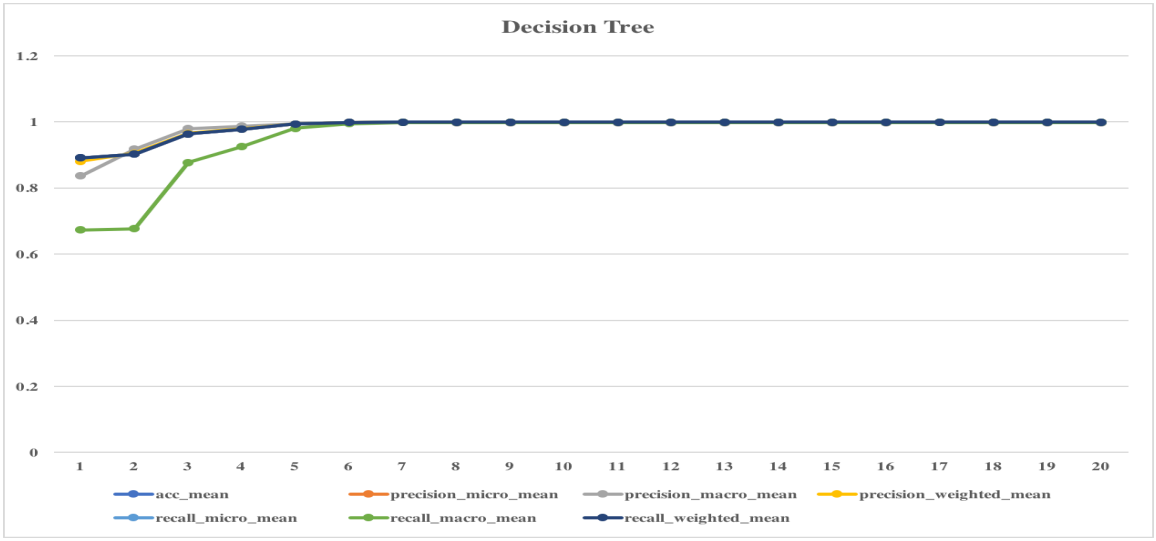


Figure 4.16. Line Graph of Decision Tree with Varying Max_Depth for Heart Disease Dataset Using Transfer Learning

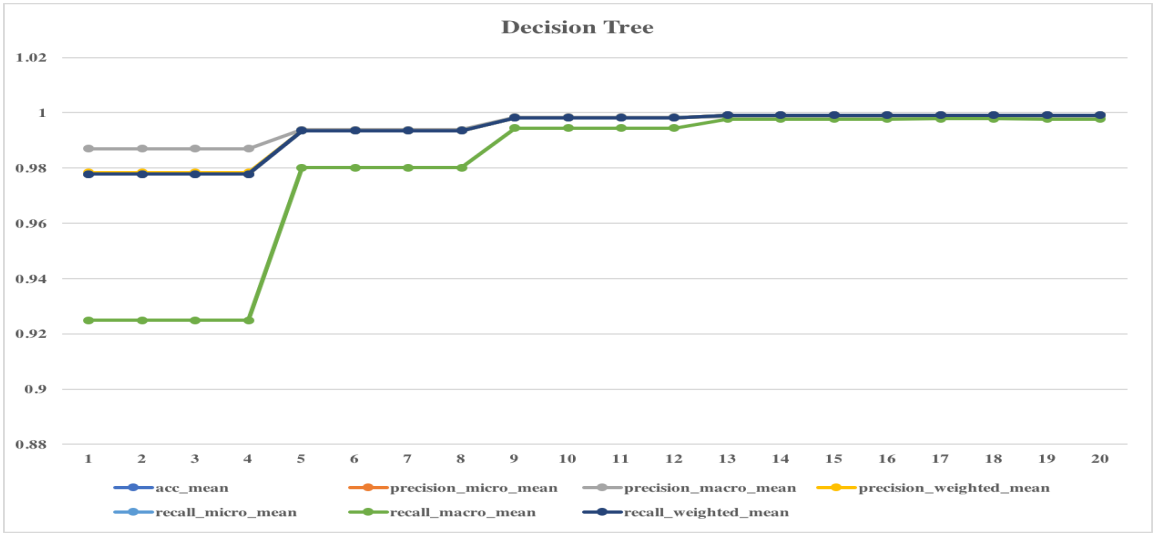


Figure 4.17. Line Graph of Decision Tree with Varying Max_Depth and Min_Sample_Split for Heart Disease Dataset Using Transfer Learning

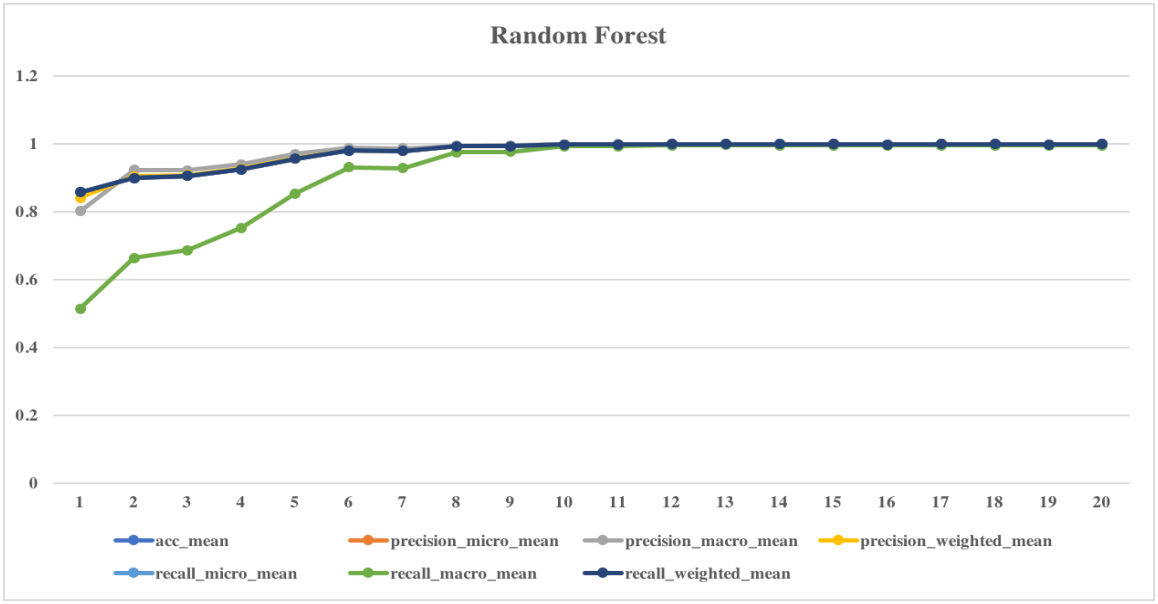


Figure 4.18. Line Graph of Random Forest with Varying Max_Depth for Heart Disease Dataset Using Transfer Learning

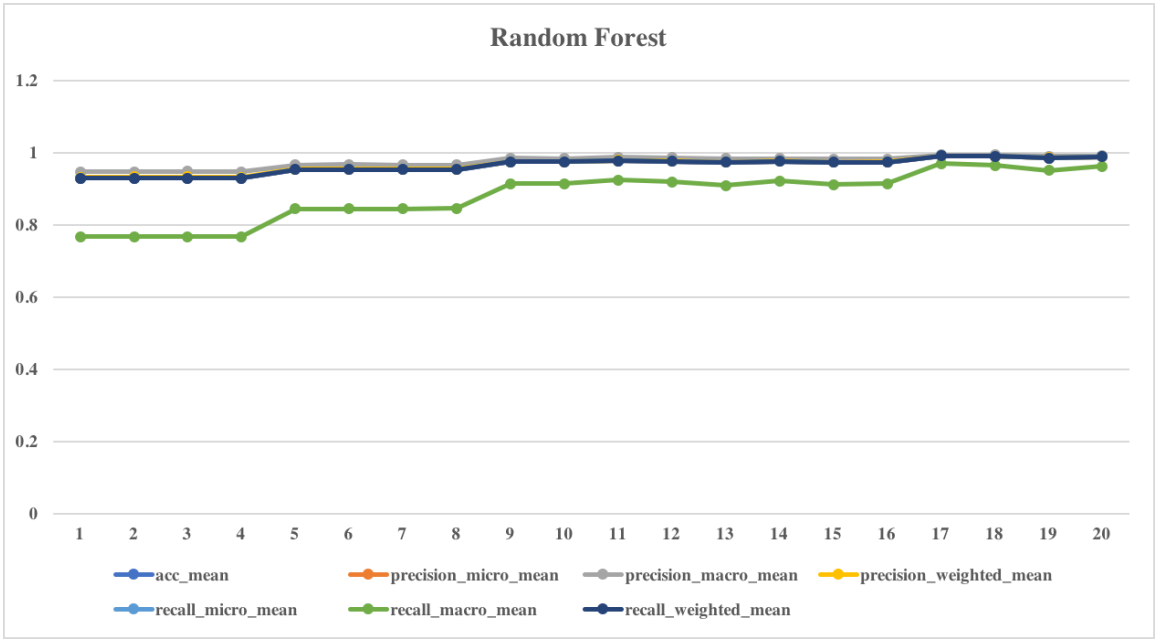


Figure 4.19. Line Graph of Random Forest with Varying Max_Depth and Min_Sample_Split for Heart Disease Dataset Using Transfer Learning

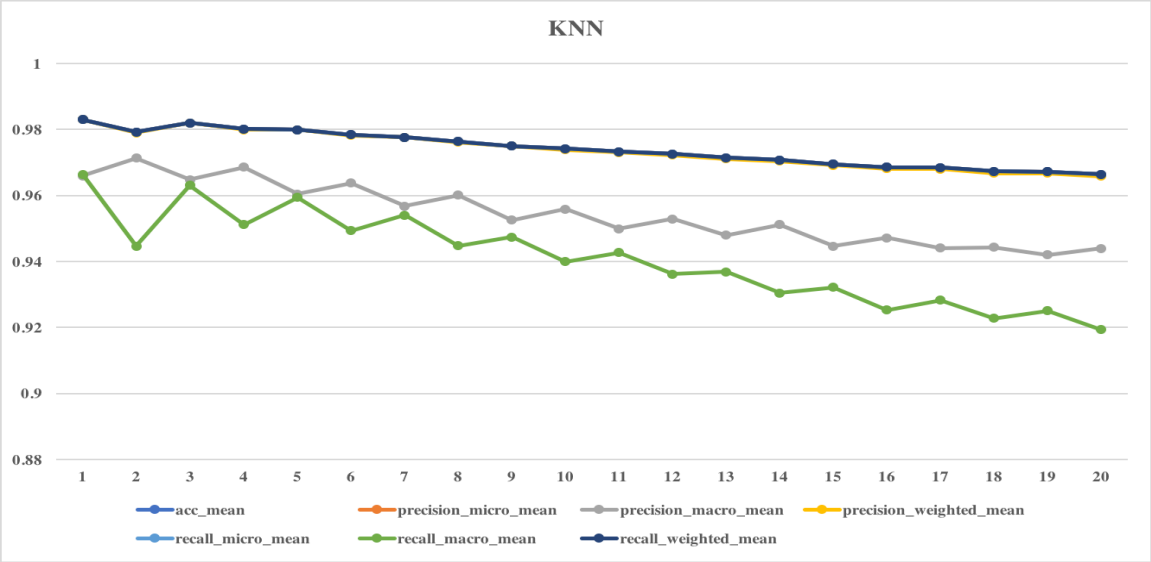


Figure 4.20. Line Graph of KNN with Varying N_Neighbor for Heart Disease Dataset Using Transfer Learning

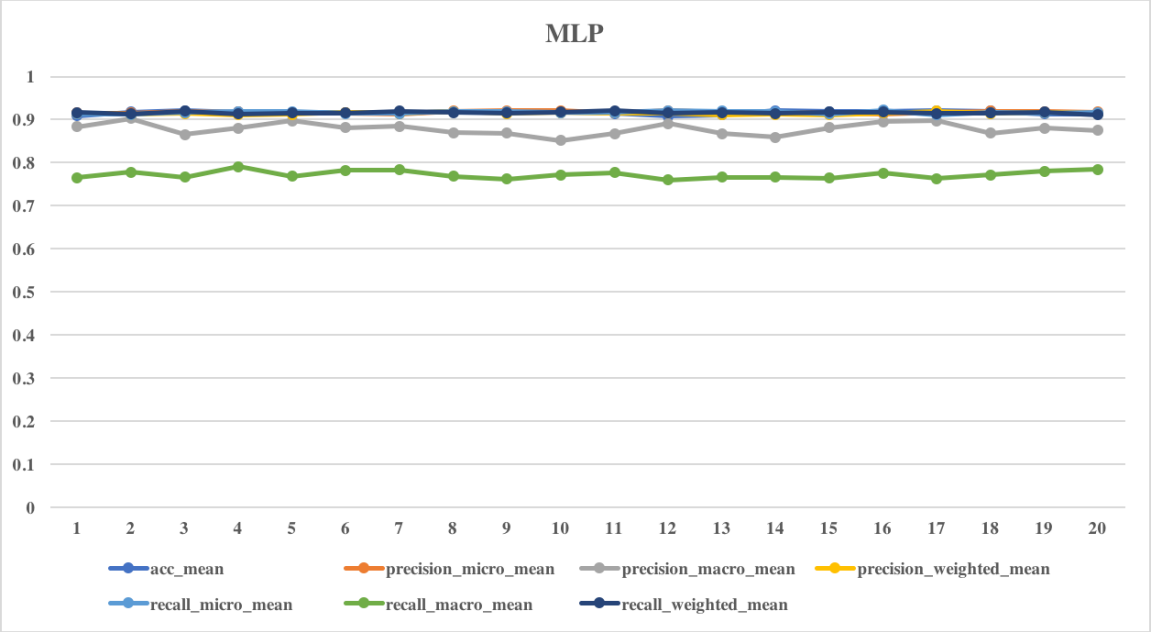


Figure 4.21. Line Graph of MLP with Varying Max_Iteration for Heart Disease Dataset Using Transfer Learning

#### 4.3.2.2 Box Plot

Following box plot shows the comparison of the evaluation metrics values for all the models created by varying max_depth of Decision Tree, max_depth of Random Forest, n_neighbor of K-Nearest Neighbor and max_iteration of MLP algorithms for heart disease dataset using transfer learning.

Table 4.8. Accuracy Value for Heart Disease Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.891079812 | 0.856703182 | 0.966405842 | 0.907929056 |
| First Quartile (Q1) | 0.996961398 | 0.972652582 | 0.969261868 | 0.914345331 |
| Median Value | 0.998904538 | 0.997600417 | 0.97370892 | 0.917214397 |
| Third Quartile(Q3) | 0.998969744 | 0.998187272 | 0.978651539 | 0.91918362 |
| Max Value | 0.999113198 | 0.998539384 | 0.982994262 | 0.921178925 |
| Box 1-hidden (Q1) | 0.996961398 | 0.972652582 | 0.969261868 | 0.914345331 |
| Box 2 (Median - Q1) | 0.00194314 | 0.024947835 | 0.004447053 | 0.002869066 |
| Box 3 (Q3- Median) | 6.52061E-05 | 0.000586854 | 0.004942619 | 0.001969223 |
| Whisker Top (Max- Q3) | 0.000143453 | 0.000352113 | 0.004342723 | 0.001995305 |
| Whisker Bottom (Q1- Min) | 0.105881586 | 0.1159494 | 0.002856025 | 0.006416275 |

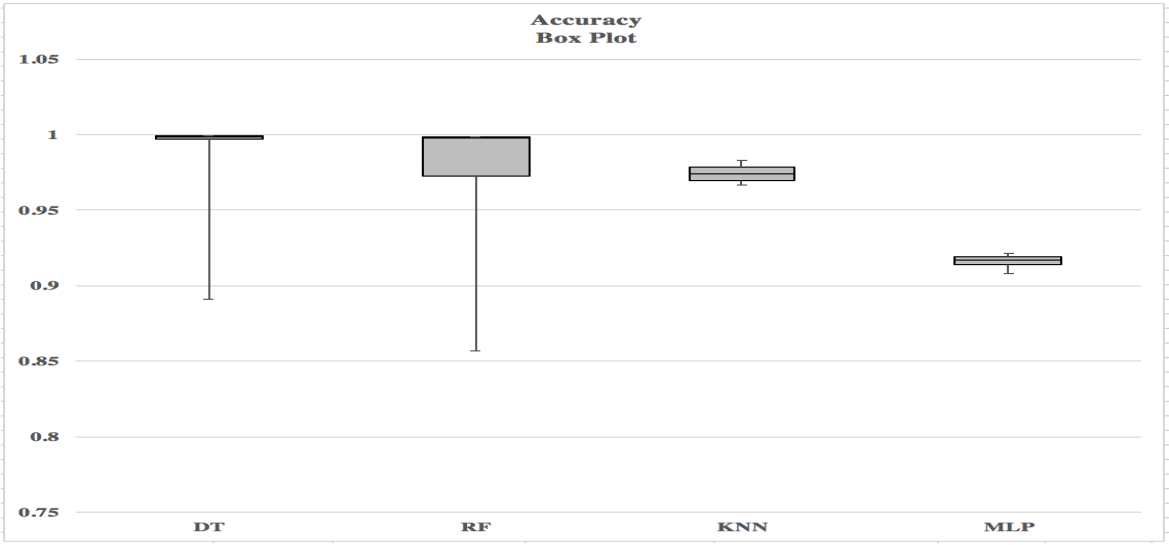


Figure 4.22. Accuracy Box Plot for Heart Disease Dataset Using Transfer Learning

Table 4.9. Precision Macro Value for Heart Disease Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.837102924 | 0.800517941 | 0.942028023 | 0.851673438 |
| First Quartile (Q1) | 0.996511645 | 0.981540284 | 0.946577899 | 0.86811231 |
| Median Value | 0.997812904 | 0.997327251 | 0.952755779 | 0.88014405 |
| Third Quartile(Q3) | 0.997991179 | 0.997824805 | 0.9613363 | 0.885949516 |
| Max Value | 0.998168142 | 0.998552782 | 0.971366674 | 0.902587897 |
| Box 1-hidden (Q1) | 0.996511645 | 0.981540284 | 0.946577899 | 0.86811231 |
| Box 2 (Median - Q1) | 0.001301259 | 0.015786966 | 0.00617788 | 0.01203174 |
| Box 3 (Q3- Median) | 0.000178276 | 0.000497554 | 0.00858052 | 0.005805466 |
| Whisker Top (Max- Q3) | 0.000176963 | 0.000727977 | 0.010030375 | 0.016638381 |
| Whisker Bottom (Q1- Min) | 0.159408721 | 0.181022343 | 0.004549877 | 0.016438872 |

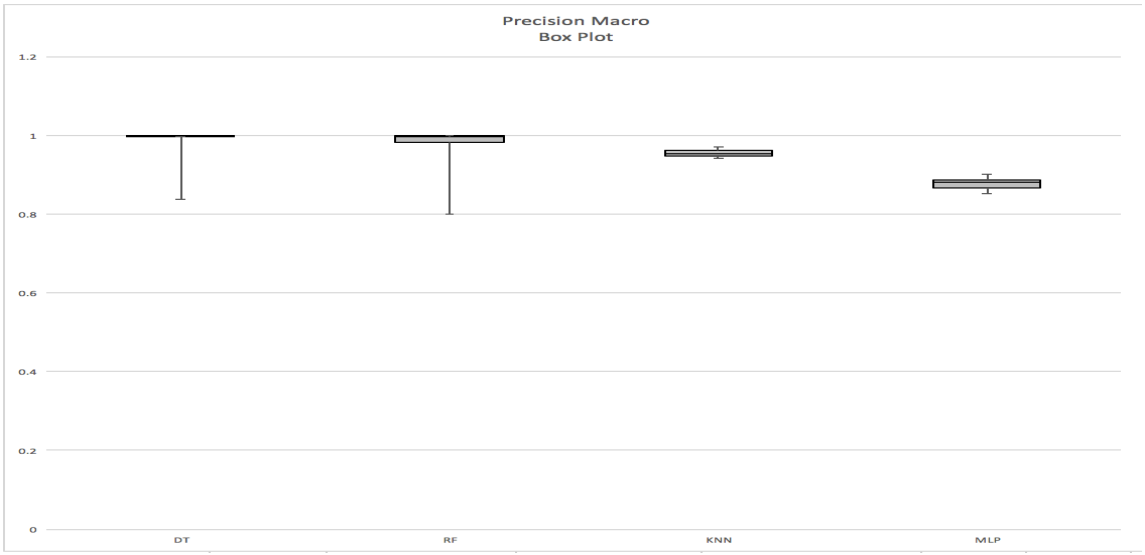


Figure 4.23. Precision Macro Box Plot for Heart Disease Dataset Using Transfer Learning

Table 4.10. Precision Micro Value for Heart Disease Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.891079812 | 0.856703182 | 0.966405842 | 0.912936881 |
| First Quartile (Q1) | 0.996961398 | 0.972652582 | 0.969261868 | 0.914906103 |
| Median Value | 0.998956703 | 0.997600417 | 0.97370892 | 0.916927491 |
| Third Quartile(Q3) | 0.999008868 | 0.998187272 | 0.978651539 | 0.918518519 |
| Max Value | 0.999061033 | 0.998539384 | 0.982994262 | 0.920918101 |
| Box 1-hidden (Q1) | 0.996961398 | 0.972652582 | 0.969261868 | 0.914906103 |
| Box 2 (Median - Q1) | 0.001995305 | 0.024947835 | 0.004447053 | 0.002021388 |
| Box 3 (Q3- Median) | 5.21648E-05 | 0.000586854 | 0.004942619 | 0.001591028 |
| Whisker Top (Max- Q3) | 5.21648E-05 | 0.000352113 | 0.004342723 | 0.002399583 |
| Whisker Bottom (Q1- Min) | 0.105881586 | 0.1159494 | 0.002856025 | 0.001969223 |

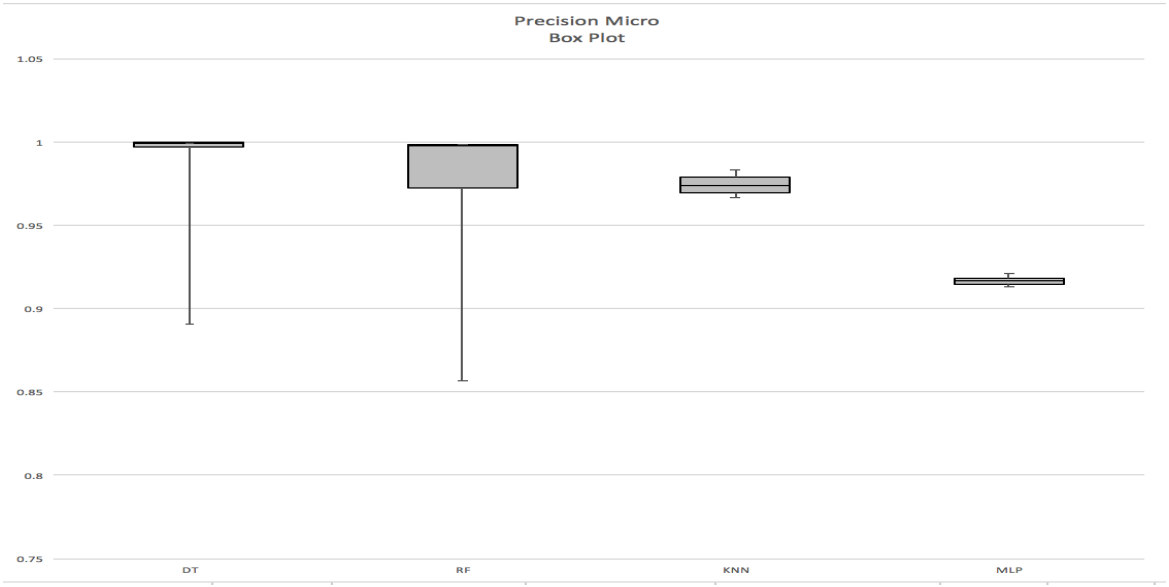


Figure 4.24. Precision Micro Box Plot for Heart Disease Dataset Using Transfer Learning

Table 4.11. Precision Weighted Value for Heart Disease Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.881309714 | 0.839906859 | 0.965824307 | 0.910897463 |
| First Quartile (Q1) | 0.996979189 | 0.973349034 | 0.96890004 | 0.912606563 |
| Median Value | 0.998959452 | 0.997605378 | 0.973513785 | 0.914109274 |
| Third Quartile(Q3) | 0.999011851 | 0.998188388 | 0.978451576 | 0.915924644 |
| Max Value | 0.999115951 | 0.998539643 | 0.983024445 | 0.919562291 |
| Box 1-hidden (Q1) | 0.996979189 | 0.973349034 | 0.96890004 | 0.912606563 |
| Box 2 (Median - Q1) | 0.001980263 | 0.024256343 | 0.004613745 | 0.001502711 |
| Box 3 (Q3-Median) | 5.23987E-05 | 0.00058301 | 0.004937791 | 0.00181537 |
| Whisker Top (Max- Q3) | 0.0001041 | 0.000351256 | 0.004572869 | 0.003637648 |
| Whisker Bottom (Q1- Min) | 0.115669475 | 0.133442176 | 0.003075733 | 0.0017091 |

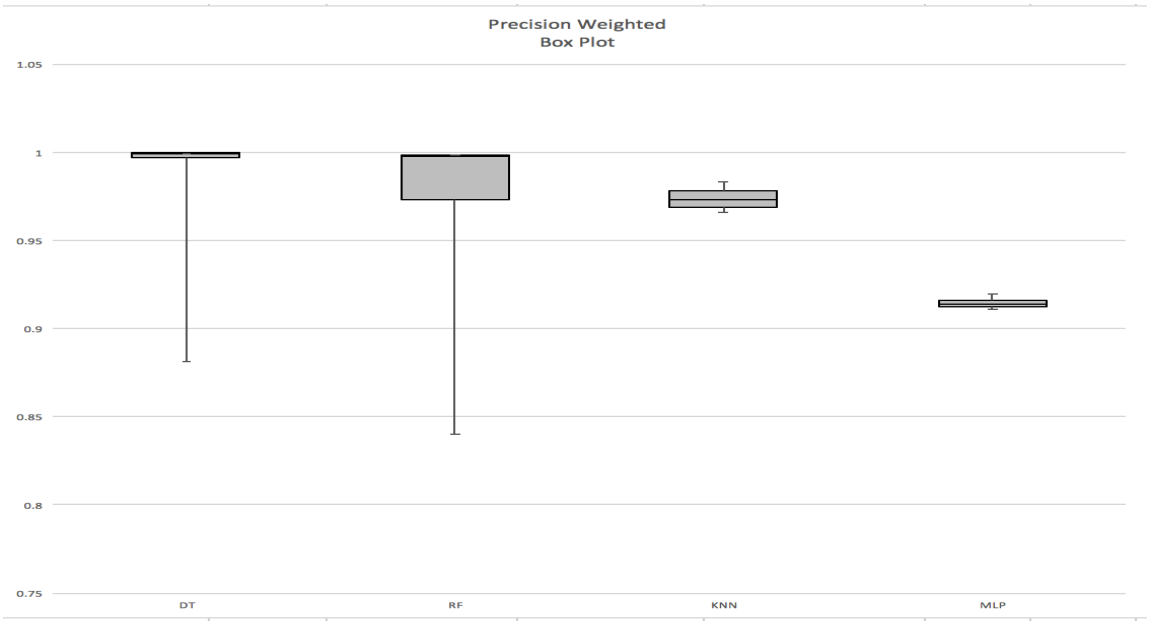


Figure 4.25. Precision Weighted Box Plot for Heart Disease Dataset Using Transfer Learning

Table 4.12. Recall Macro Value for Heart Disease Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.673276336 | 0.513183911 | 0.919357294 | 0.760213714 |
| First Quartile (Q1) | 0.990946322 | 0.909174728 | 0.929915219 | 0.765635212 |
| Median Value | 0.998198116 | 0.992790826 | 0.941333617 | 0.770014251 |
| Third Quartile(Q3) | 0.998243978 | 0.994988066 | 0.949797825 | 0.778264013 |
| Max Value | 0.998305135 | 0.995619396 | 0.966387175 | 0.790494144 |
| Box 1-hidden (Q1) | 0.990946322 | 0.909174728 | 0.929915219 | 0.765635212 |
| Box 2 (Median - Q1) | 0.007251795 | 0.083616098 | 0.011418398 | 0.00437904 |
| Box 3 (Q3- Median) | 4.58612E-05 | 0.00219724 | 0.008464208 | 0.008249762 |
| Whisker Top (Max- Q3) | 6.11577E-05 | 0.00063133 | 0.016589351 | 0.012230131 |
| Whisker Bottom (Q1- Min) | 0.317669986 | 0.395990817 | 0.010557925 | 0.005421498 |

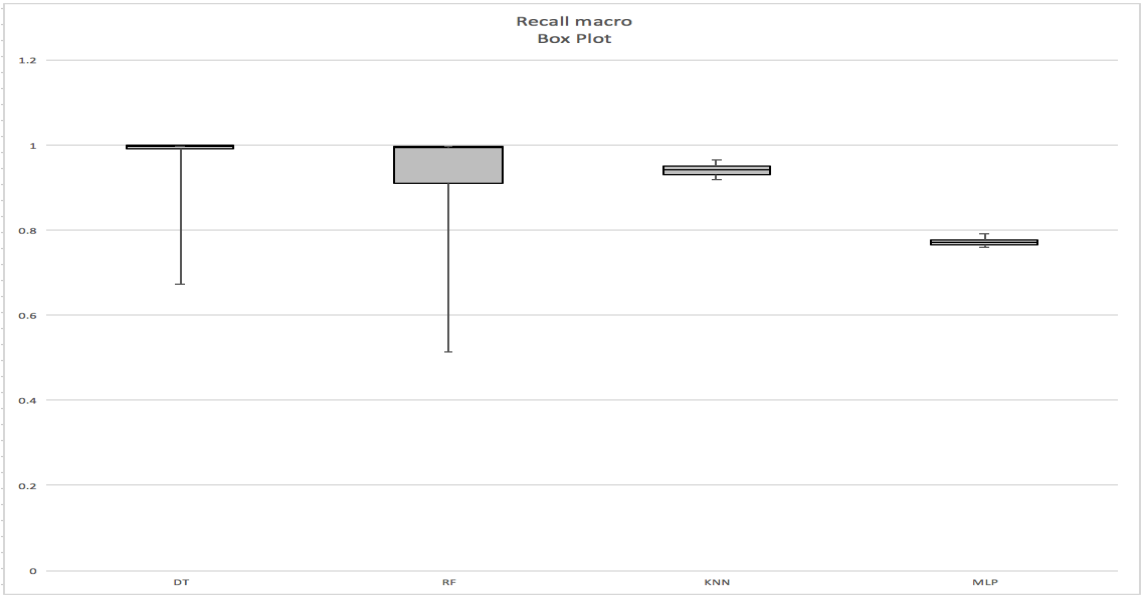


Figure 4.26. Recall Macro Box Plot for Heart Disease Dataset Using Transfer Learning

Table 4.13. Recall Micro Value for Heart Disease Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.891079812 | 0.856703182 | 0.965824307 | 0.911371935 |
| First Quartile (Q1) | 0.996961398 | 0.972652582 | 0.96890004 | 0.91422796 |
| Median Value | 0.998982786 | 0.997600417 | 0.973513785 | 0.916588419 |
| Third Quartile(Q3) | 0.999061033 | 0.998187272 | 0.978451576 | 0.918935837 |
| Max Value | 0.999061033 | 0.998539384 | 0.983024445 | 0.922065728 |
| Box 1-hidden (Q1) | 0.996961398 | 0.972652582 | 0.96890004 | 0.91422796 |
| Box 2 (Median - Q1) | 0.002021388 | 0.024947835 | 0.004613745 | 0.002360459 |
| Box 3 (Q3- Median) | 7.82473E-05 | 0.000586854 | 0.004937791 | 0.002347418 |
| Whisker Top (Max- Q3) | 0 | 0.000352113 | 0.004572869 | 0.00312989 |
| Whisker Bottom (Q1- Min) | 0.105881586 | 0.1159494 | 0.003075733 | 0.002856025 |

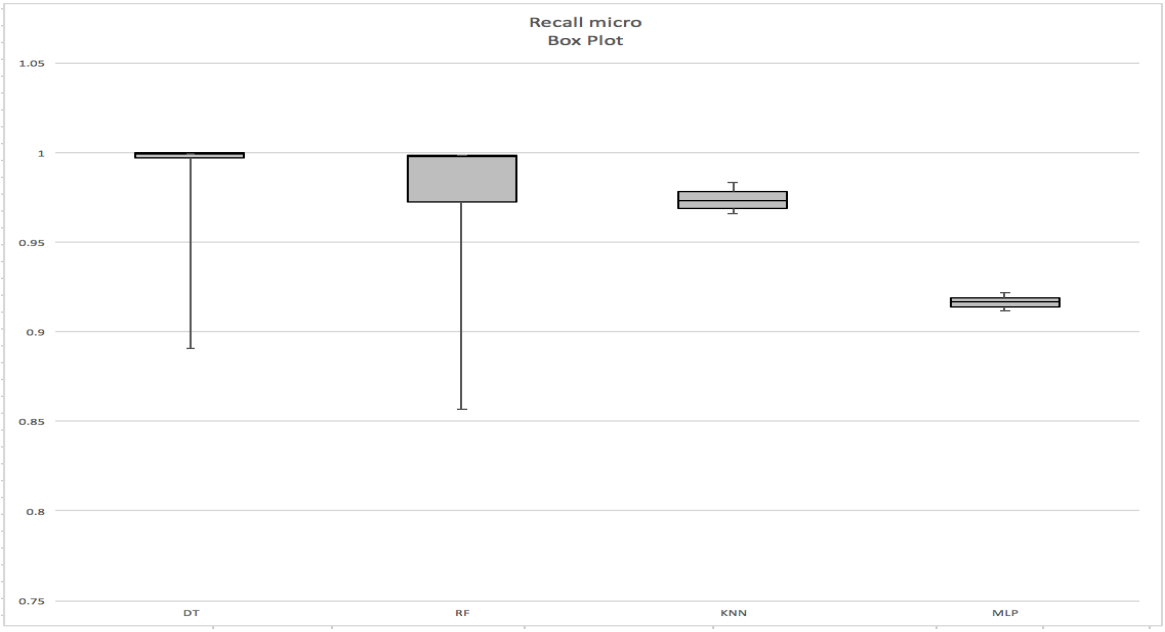


Figure 4.27. Recall Micro Box Plot for Heart Disease Dataset Using Transfer Learning

Table 4.14. Recall Weighted Value for Heart Disease Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.891079812 | 0.856703182 | 0.966405842 | 0.91131977 |
| First Quartile (Q1) | 0.996922274 | 0.972652582 | 0.969261868 | 0.914814815 |
| Median Value | 0.998956703 | 0.997600417 | 0.97370892 | 0.916197183 |
| Third Quartile(Q3) | 0.999008868 | 0.998187272 | 0.978651539 | 0.917318727 |
| Max Value | 0.999113198 | 0.998539384 | 0.982994262 | 0.920344288 |
| Box 1-hidden (Q1) | 0.996922274 | 0.972652582 | 0.969261868 | 0.914814815 |
| Box 2 (Median - Q1) | 0.002034429 | 0.024947835 | 0.004447053 | 0.001382368 |
| Box 3 (Q3- Median) | 5.21648E-05 | 0.000586854 | 0.004942619 | 0.001121544 |
| Whisker Top (Max- Q3) | 0.00010433 | 0.000352113 | 0.004342723 | 0.003025561 |
| Whisker Bottom (Q1- Min) | 0.105842462 | 0.1159494 | 0.002856025 | 0.003495044 |

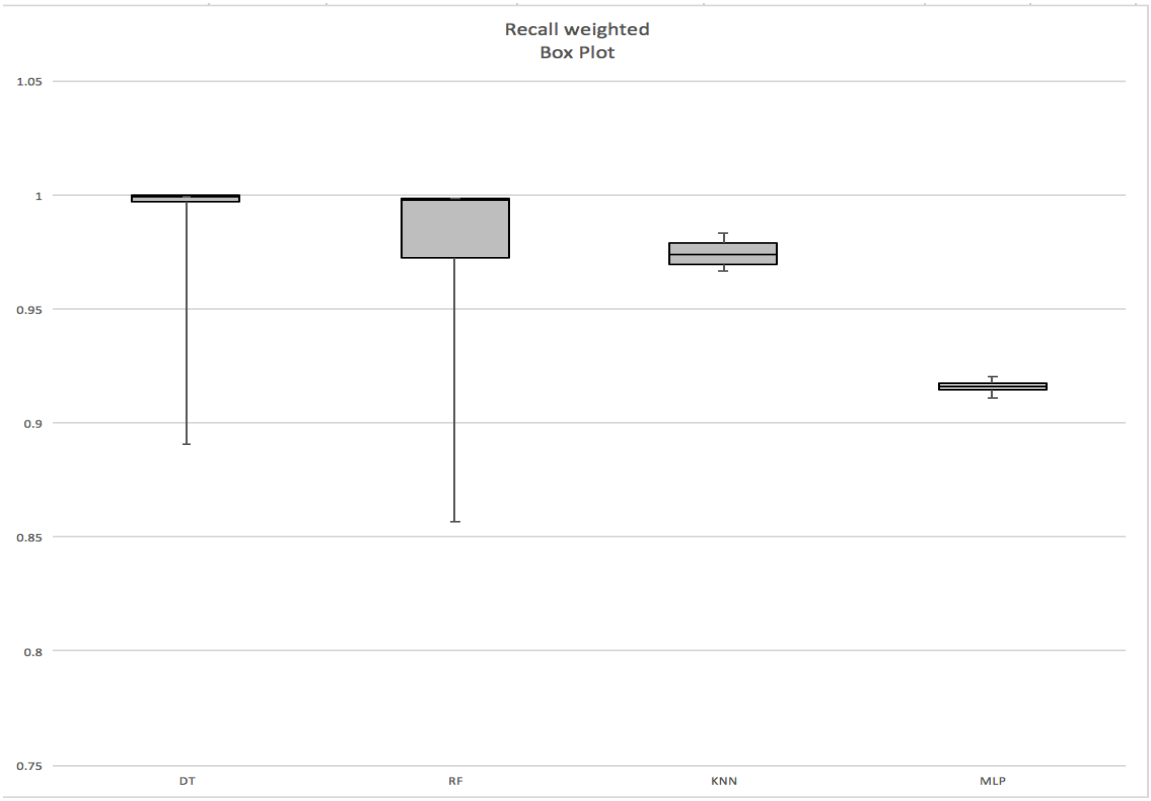


Figure 4.28. Recall Weighted Box Plot for Heart Disease Dataset Using Transfer Learning

#### 4.3.2.3 Best Model

The following diagram shows the best model of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithm for each evaluation metrics.

| Algorithm | Accuracy | | Precision Micro | | Precision Macro | | Precision Weighted | | Recall Micro | | Recall Macro | | Recall Weighted | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value |
| Decission Tree | max_depth: 18 | 0.999113198 | max_depth: 7 | 0.999061033 | max_depth: 11 | 0.998168142 | max_depth: 9 | 0.999115951 | max_depth: 7 | 0.999061033 | max_depth: 9 | 0.998305135 | max_depth: 13 | 0.999113198 |
| Random Forest | max_depth: 13 | 0.998539384 | max_depth: 13 | 0.998539384 | max_depth: 13 | 0.998552782 | max_depth: 13 | 0.998539643 | max_depth: 13 | 0.998539384 | max_depth: 13 | 0.995619396 | max_depth: 13 | 0.998539384 |
| KNN | n_neighbors: 1 | 0.982994262 | n_neighbors: 1 | 0.982994262 | n_neighbors: 2 | 0.971366674 | n_neighbors: 1 | 0.983024445 | n_neighbors: 1 | 0.982994262 | n_neighbors: 1 | 0.966387175 | n_neighbors: 1 | 0.982994262 |
| MLP | max_iter: 140000 | 0.921178925 | max_iter: 90000 | 0.920918101 | max_iter: 20000 | 0.902587897 | max_iter: 170000 | 0.919562291 | max_iter: 160000 | 0.922065728 | max_iter: 40000 | 0.790494144 | max_iter: 110000 | 0.920344288 |

Figure 4.29. Best Models for Heart Disease Dataset Using Transfer Learning

#### 4.3.2.4 ROC

Here we compared the ROC curve of the best models of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithm as follows:

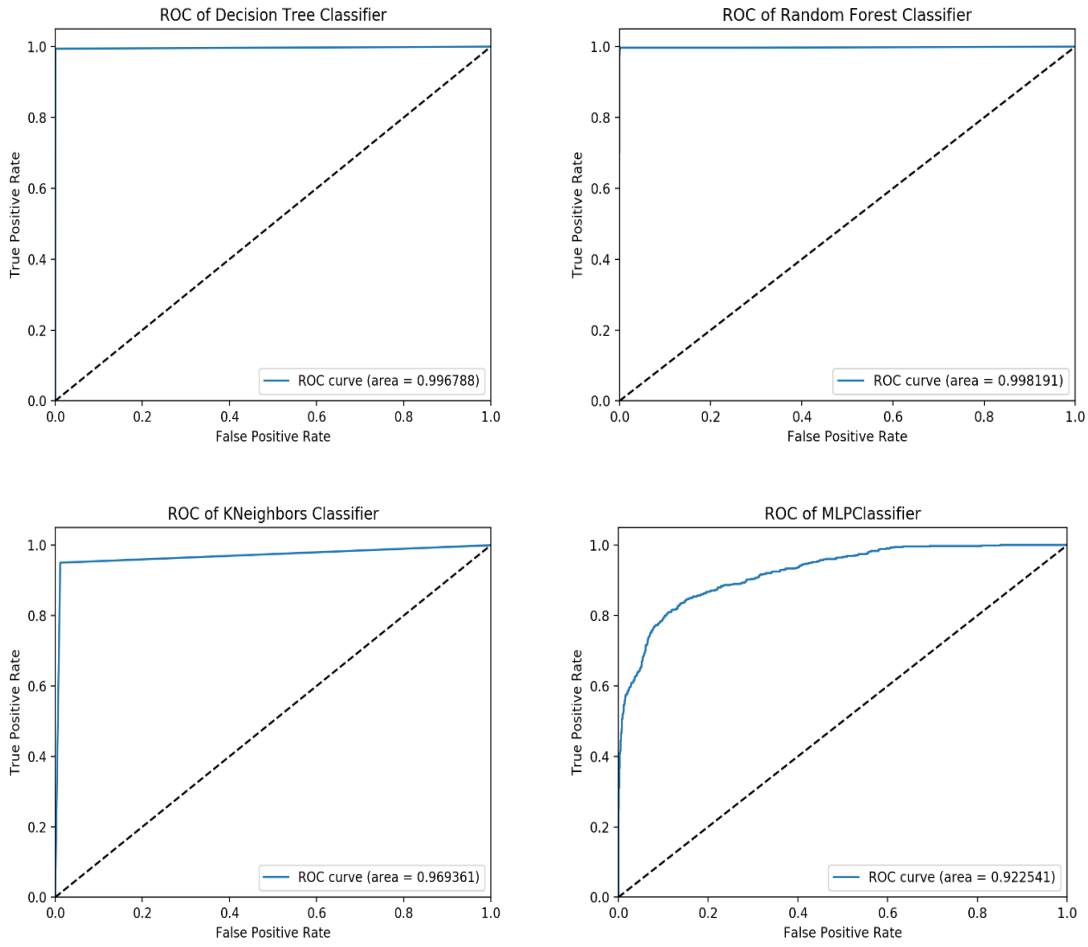


Figure 4.30. ROC Curve of Best Models for Heart Disease Dataset Using Transfer Learning

### 4.3.3 Using Suggested Features

For this experiment, experts identified 11 important features out of heart disease dataset as shown in section 3.3. We used these identified important features for training all the models of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithm and estimated the

evaluation metrics for each model using grid search 5-fold cross validation. Following sections shows the comparison between the models.

#### 4.3.3.1 Line Graph

Following line diagram shows the comparison of different models of each algorithm based on evaluation metrics.

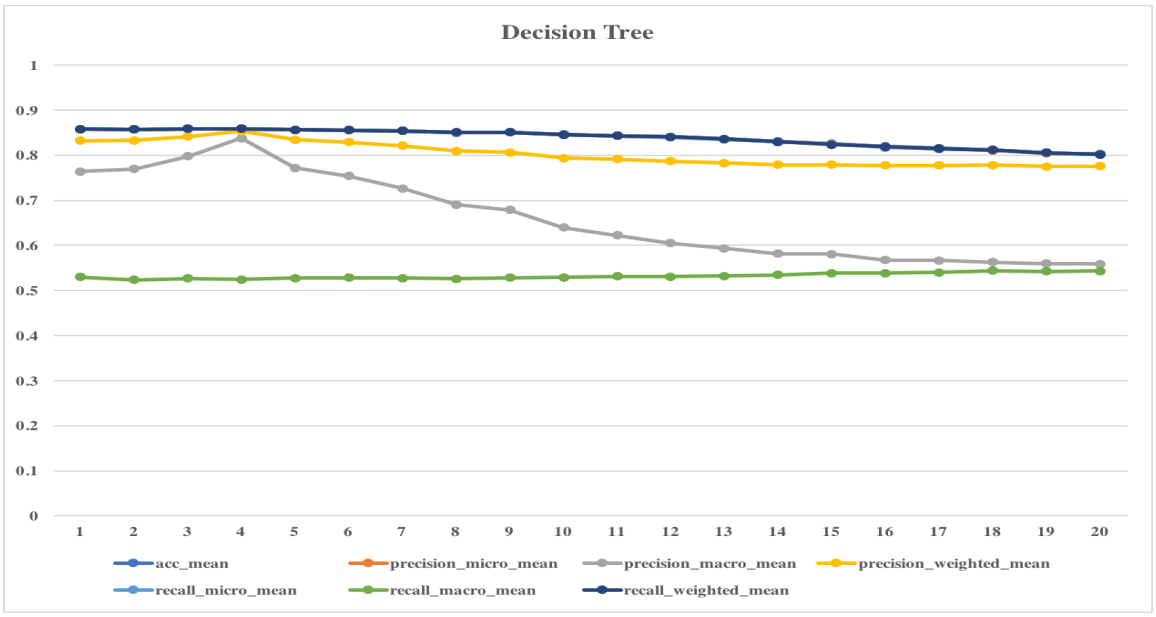


Figure 4.31. Line Graph of Decision Tree with Varying Max_Depth for Heart Disease Dataset Using Suggested Features

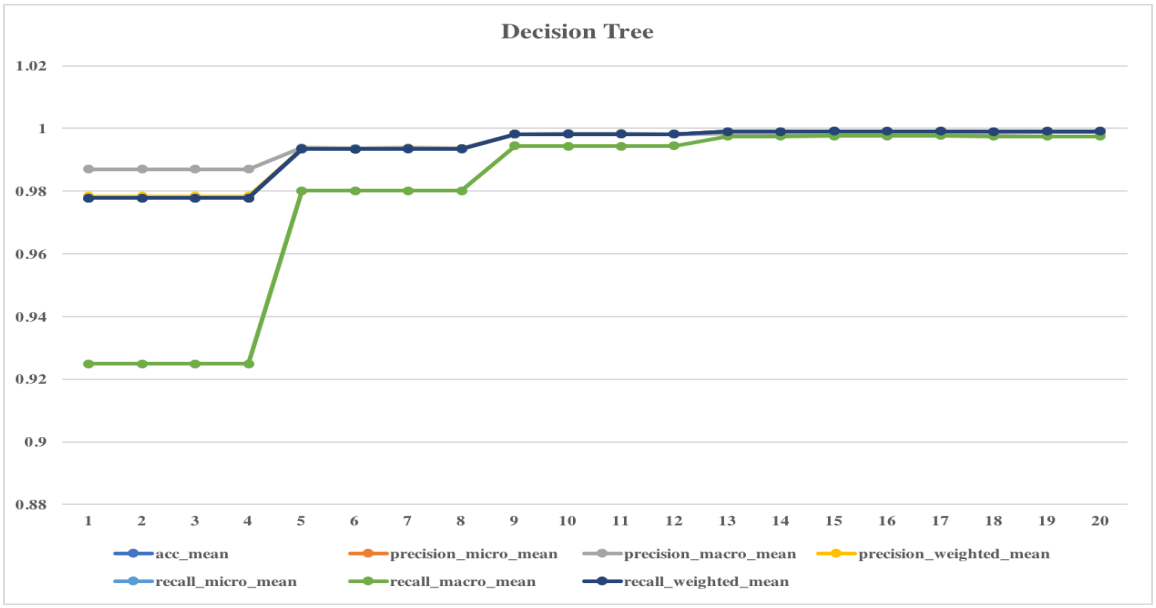


Figure 4.32. Line Graph of Decision Tree with Varying Max_Depth and Min_Sample_Split for Heart Disease Dataset Using Suggested Features

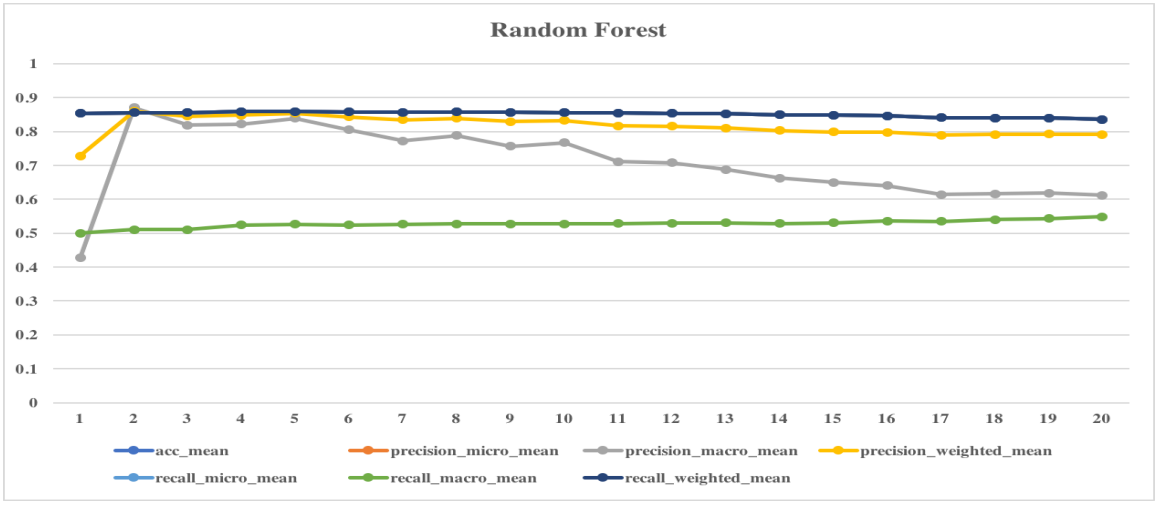


Figure 4.33. Line Graph of Decision Tree with Varying Max_Depth for Heart Disease Dataset Using Suggested Features.

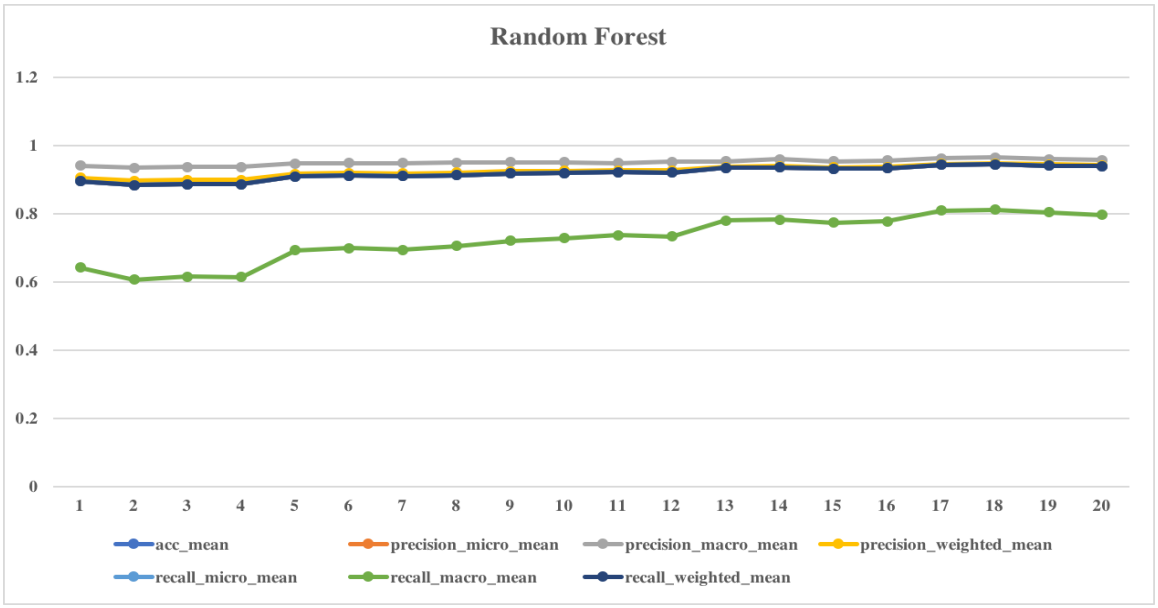


Figure 4.34. Line Graph of Random Forest with Varying Max_Depth and Min_Sample_Split for Heart Disease Dataset Using Suggested Features.

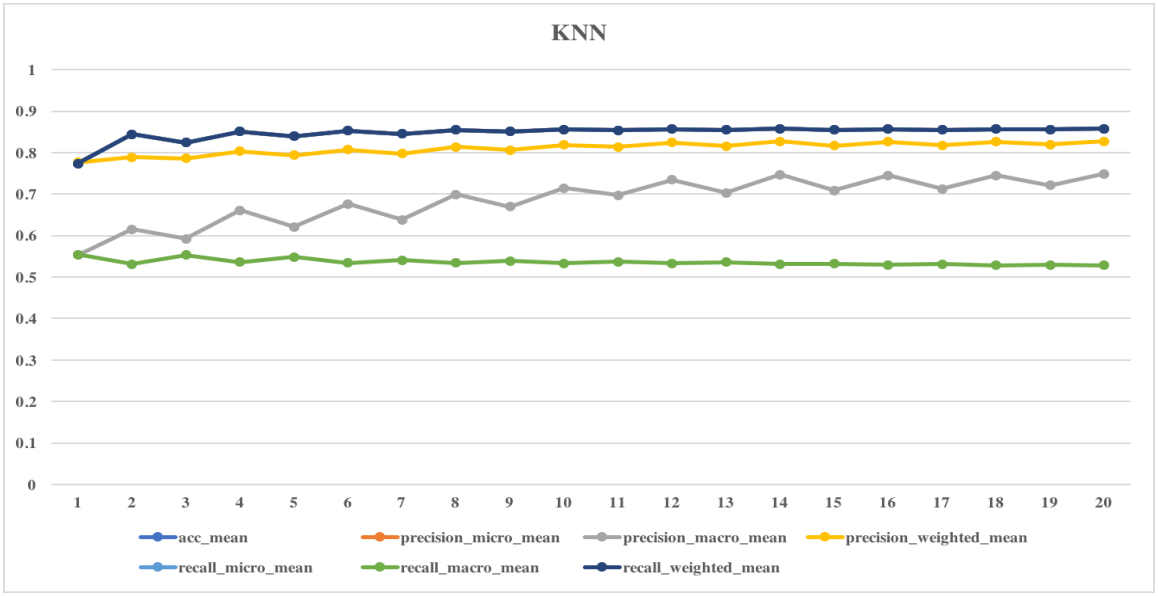


Figure 4.35. Line Graph of KNN with Varying N_Neighbor for Heart Disease Dataset Using Suggested Features

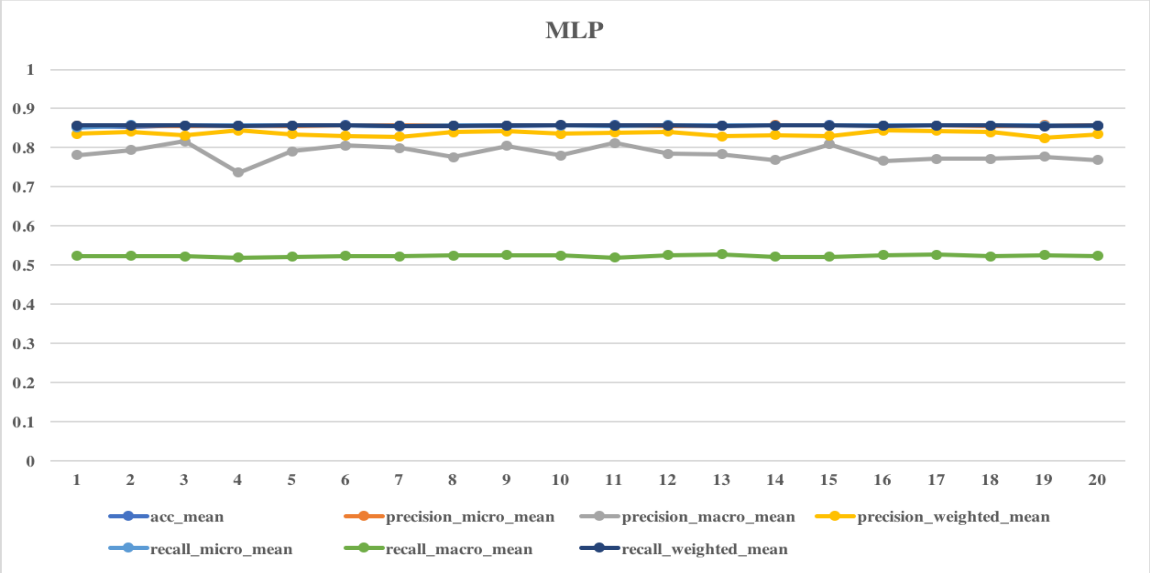


Figure 4.36. Line Graph of MLP with Varying Max_Iteration for Heart Disease Dataset Using Suggested Features

#### 4.3.3.2 Box Plot

Following box plot shows the comparison of the evaluation metrics values for all the models created by varying max_depth of Decision Tree, max_depth of Random Forest, n_neighbor of K-Nearest Neighbor and max_iteration of MLP algorithms for heart disease dataset using only suggested features.

Table 4.15. Accuracy Weighted Value for Heart Disease Dataset Using Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.801877934 | 0.835837246 | 0.773291601 | 0.853834116 |
| First Quartile (Q1) | 0.823878456 | 0.847548252 | 0.849308816 | 0.856442358 |
| Median Value | 0.844522692 | 0.853781951 | 0.854512259 | 0.856781429 |
| Third Quartile(Q3) | 0.855581638 | 0.856285863 | 0.856181534 | 0.857042254 |
| Max Value | 0.858581116 | 0.8590506 | 0.85722483 | 0.857642149 |
| Box 1-hidden (Q1) | 0.823878456 | 0.847548252 | 0.849308816 | 0.856442358 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Box 2 (Median - Q1) | 0.020644236 | 0.006233698 | 0.005203443 | 0.000339071 |
| Box 3 (Q3- Median) | 0.011058946 | 0.002503912 | 0.001669275 | 0.000260824 |
| Whisker Top (Max- Q3) | 0.002999478 | 0.002764737 | 0.001043297 | 0.000599896 |
| Whisker Bottom (Q1- Min) | 0.022000522 | 0.011711007 | 0.076017214 | 0.002608242 |

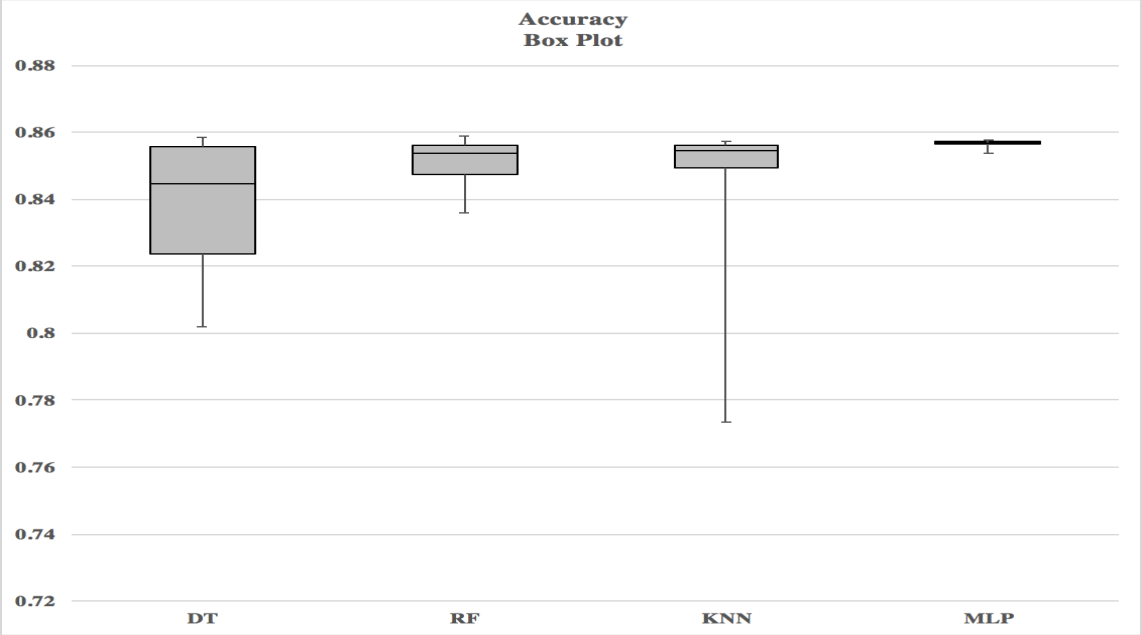


Figure 4.37. Accuracy Box Plot for Heart Disease Dataset Using Suggested Features

Table 4.16. Precision Macro Value for Heart Disease Dataset Using Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.558728838 | 0.426499739 | 0.552827739 | 0.736792799 |
| First Quartile (Q1) | 0.577150124 | 0.634896059 | 0.655364576 | 0.771711809 |
| Median Value | 0.630733801 | 0.709333602 | 0.701272512 | 0.782239266 |
| Third Quartile(Q3) | 0.756240757 | 0.791830576 | 0.724351373 | 0.800412557 |
| Max Value | 0.837718605 | 0.86990325 | 0.748469484 | 0.816266373 |
| Box 1-hidden (Q1) | 0.577150124 | 0.634896059 | 0.655364576 | 0.771711809 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Box 2 (Median - Q1) | 0.053583677 | 0.074437543 | 0.045907936 | 0.010527456 |
| Box 3 (Q3- Median) | 0.125506956 | 0.082496974 | 0.023078861 | 0.018173291 |
| Whisker Top (Max- Q3) | 0.081477848 | 0.078072674 | 0.024118111 | 0.015853817 |
| Whisker Bottom (Q1- Min) | 0.018421286 | 0.20839632 | 0.102536837 | 0.034919011 |

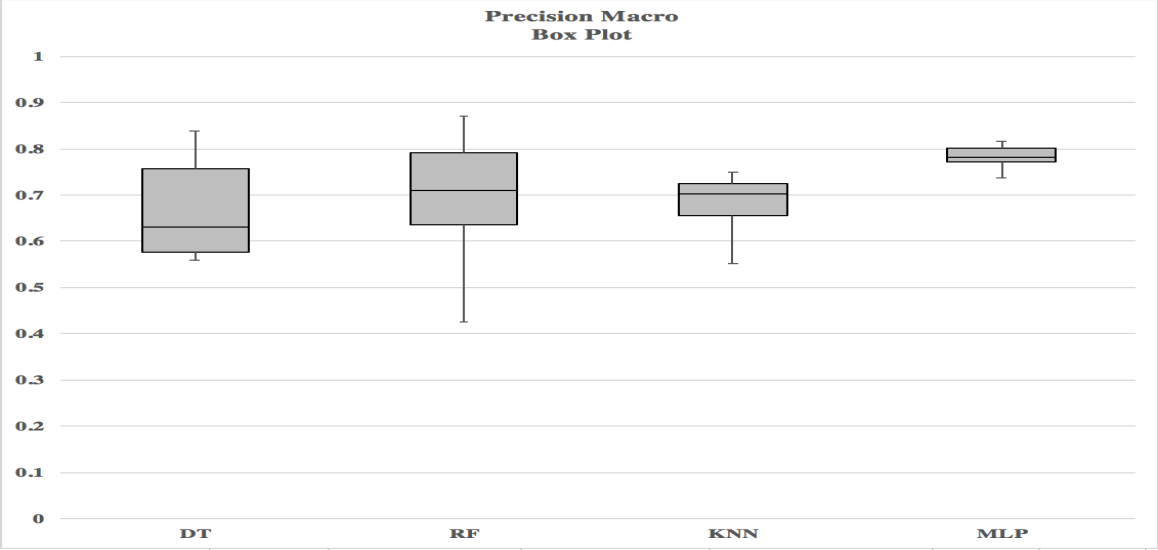


Figure 4.38. Precision Macro Box Plot for Heart Disease Dataset Using Suggested Features

Table 4.17. Precision Micro Value for Heart Disease Dataset Using Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.80312989 | 0.835837246 | 0.773291601 | 0.855033907 |
| First Quartile (Q1) | 0.824100156 | 0.847548252 | 0.849308816 | 0.856377152 |
| Median Value | 0.844418362 | 0.853781951 | 0.854512259 | 0.856651017 |
| Third Quartile(Q3) | 0.855738132 | 0.856285863 | 0.856181534 | 0.857016171 |
| Max Value | 0.858581116 | 0.8590506 | 0.85722483 | 0.857746479 |
| Box 1-hidden (Q1) | 0.824100156 | 0.847548252 | 0.849308816 | 0.856377152 |
| Box 2 (Median - Q1) | 0.020318206 | 0.006233698 | 0.005203443 | 0.000273865 |
| Box 3 (Q3- Median) | 0.01131977 | 0.002503912 | 0.001669275 | 0.000365154 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Whisker Top (Max- Q3) | 0.002842984 | 0.002764737 | 0.001043297 | 0.000730308 |
| Whisker Bottom (Q1- Min) | 0.020970266 | 0.011711007 | 0.076017214 | 0.001343245 |

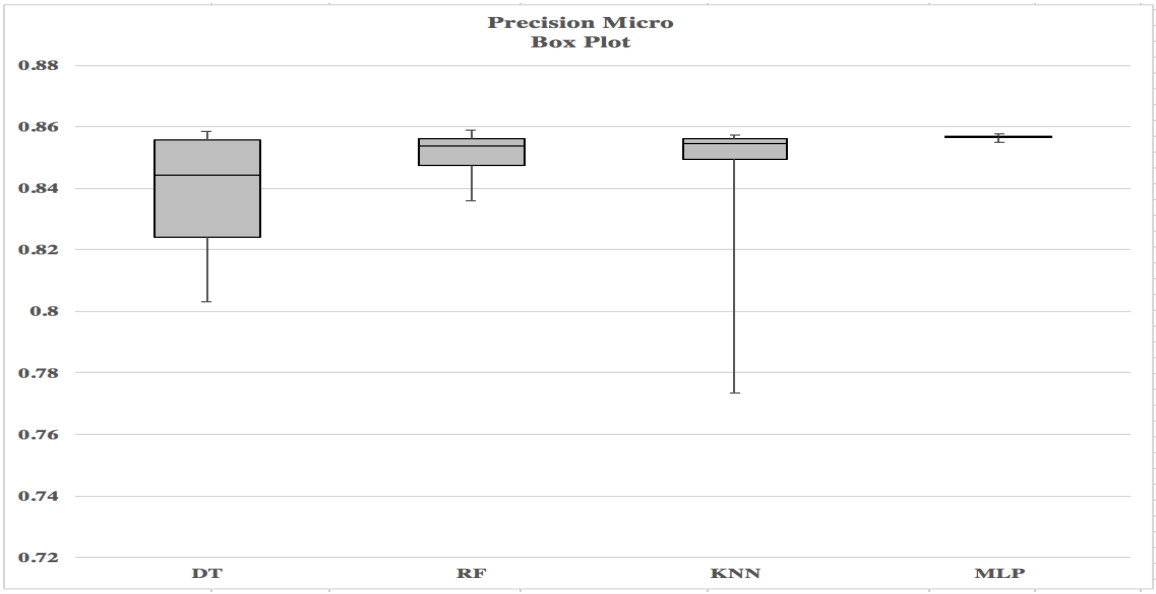


Figure 4.39. Precision Micro Box Plot for Heart Disease Dataset Using Suggested Features

Table 4.18. Precision Weighted Value for Heart Disease Dataset Using Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.775145221 | 0.72760812 | 0.776143265 | 0.824905059 |
| First Quartile (Q1) | 0.77855232 | 0.795869977 | 0.801699973 | 0.831141034 |
| Median Value | 0.792295358 | 0.815996177 | 0.814758881 | 0.835691648 |
| Third Quartile(Q3) | 0.829560231 | 0.839812378 | 0.820681264 | 0.841087238 |
| Max Value | 0.852828512 | 0.859760048 | 0.827417288 | 0.844376754 |
| Box 1-hidden (Q1) | 0.77855232 | 0.795869977 | 0.801699973 | 0.831141034 |
| Box 2 (Median - Q1) | 0.013743039 | 0.020126201 | 0.013058908 | 0.004550614 |
| Box 3 (Q3- Median) | 0.037264873 | 0.0238162 | 0.005922383 | 0.005395589 |
| Whisker Top (Max- Q3) | 0.02326828 | 0.01994767 | 0.006736024 | 0.003289516 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Whisker Bottom (Q1- Min) | 0.003407099 | 0.068261857 | 0.025556708 | 0.006235976 |

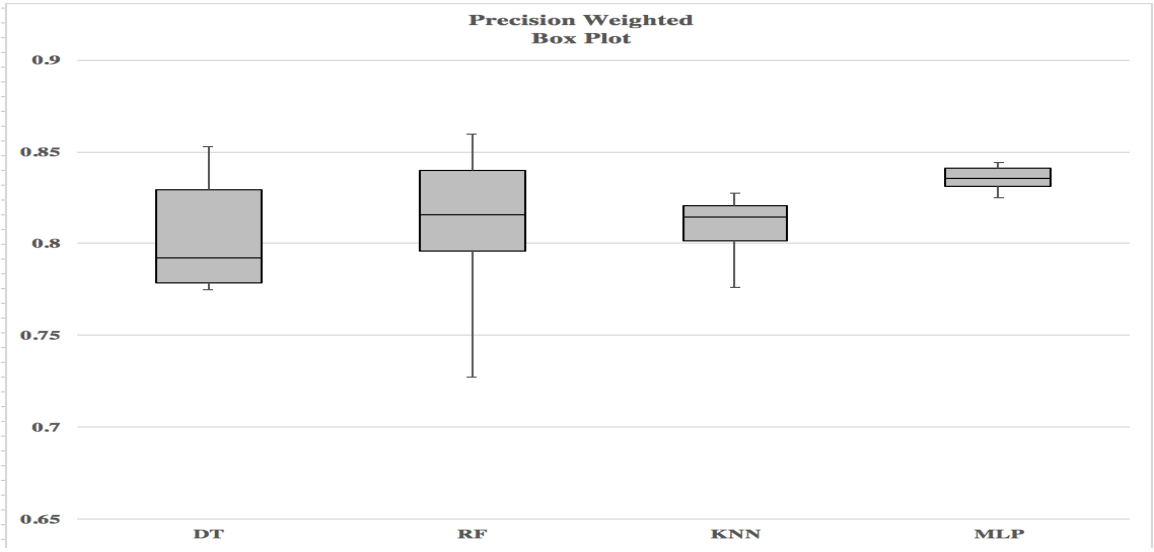


Figure 4.40. Precision Weighted Box Plot for Heart Disease Dataset Using Suggested Features

Table 4.19. Recall Macro Value for Heart Disease Dataset Using Suggested Features.

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.523699028 | 0.5 | 0.528544589 | 0.51911651 |
| First Quartile (Q1) | 0.527386891 | 0.525474858 | 0.530920784 | 0.52199745 |
| Median Value | 0.529857476 | 0.528032552 | 0.533498584 | 0.523624214 |
| Third Quartile(Q3) | 0.538357635 | 0.531585862 | 0.537458344 | 0.525324117 |
| Max Value | 0.543696941 | 0.548233019 | 0.554452224 | 0.52766627 |
| Box 1-hidden (Q1) | 0.527386891 | 0.525474858 | 0.530920784 | 0.52199745 |
| Box 2 (Median - Q1) | 0.002470585 | 0.002557695 | 0.0025778 | 0.001626764 |
| Box 3 (Q3- Median) | 0.008500159 | 0.00355331 | 0.00395976 | 0.001699904 |
| Whisker Top (Max- Q3) | 0.005339306 | 0.016647157 | 0.01699388 | 0.002342153 |
| Whisker Bottom (Q1- Min) | 0.003687863 | 0.025474858 | 0.002376196 | 0.00288094 |

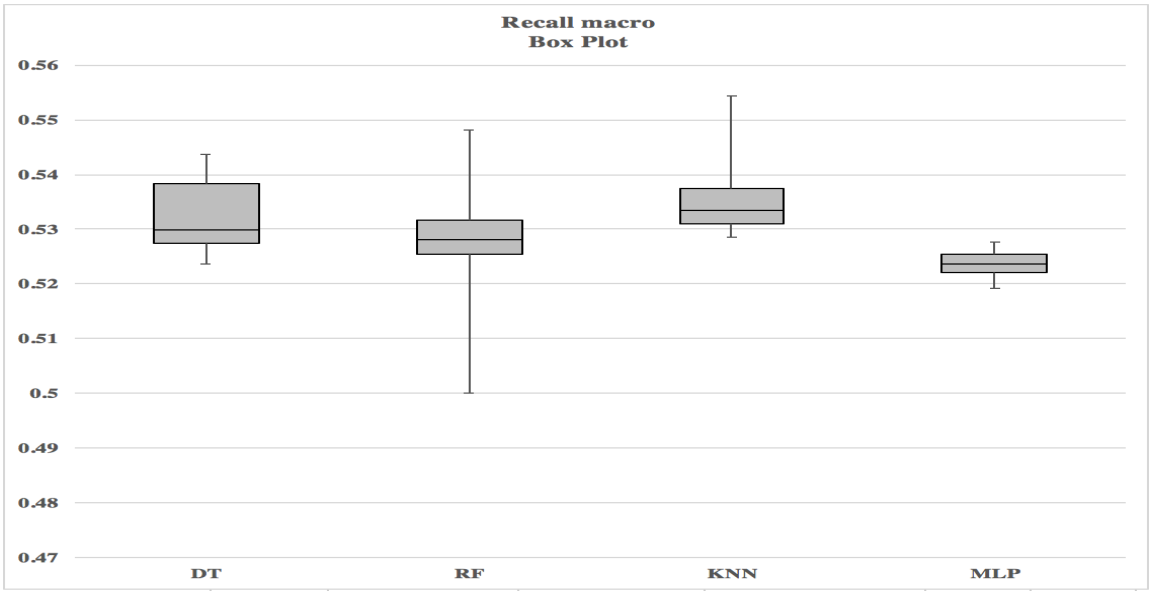


Figure 4.41. Recall Macro Box Plot for Heart Disease Dataset Using Suggested Features

Table 4.20. Recall Micro Value for Heart Disease Dataset Using Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.802034429 | 0.835837246 | 0.773291601 | 0.851017214 |
| First Quartile (Q1) | 0.823004695 | 0.847548252 | 0.849308816 | 0.856664058 |
| Median Value | 0.844653104 | 0.853781951 | 0.854512259 | 0.856833594 |
| Third Quartile(Q3) | 0.85571205 | 0.856285863 | 0.856181534 | 0.857133542 |
| Max Value | 0.858581116 | 0.8590506 | 0.85722483 | 0.857850809 |
| Box 1-hidden (Q1) | 0.823004695 | 0.847548252 | 0.849308816 | 0.856664058 |
| Box 2 (Median - Q1) | 0.021648409 | 0.006233698 | 0.005203443 | 0.000169536 |
| Box 3 (Q3-Median) | 0.011058946 | 0.002503912 | 0.001669275 | 0.000299948 |
| Whisker Top (Max- Q3) | 0.002869066 | 0.002764737 | 0.001043297 | 0.000717267 |
| Whisker Bottom (Q1- Min) | 0.020970266 | 0.011711007 | 0.076017214 | 0.005646844 |

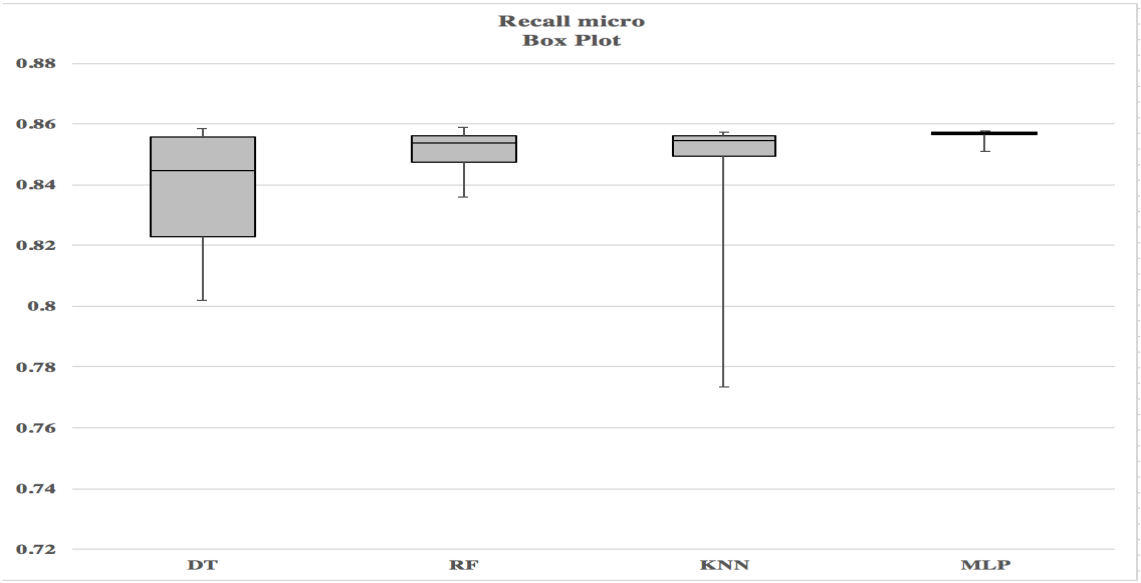


Figure 4.42. Recall Micro Box Plot for Heart Disease Dataset Using Suggested Features

Table 4.21. Recall Weight Value for Heart Disease Dataset Using Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.801408451 | 0.835837246 | 0.773291601 | 0.854773083 |
| First Quartile (Q1) | 0.822130934 | 0.847548252 | 0.849308816 | 0.856272822 |
| Median Value | 0.844209703 | 0.853781951 | 0.854512259 | 0.856755347 |
| Third Quartile(Q3) | 0.855568597 | 0.856285863 | 0.856181534 | 0.857146583 |
| Max Value | 0.858581116 | 0.8590506 | 0.85722483 | 0.857902973 |
| Box 1-hidden (Q1) | 0.822130934 | 0.847548252 | 0.849308816 | 0.856272822 |
| Box 2 (Median - Q1) | 0.022078769 | 0.006233698 | 0.005203443 | 0.000482525 |
| Box 3 (Q3-Median) | 0.011358894 | 0.002503912 | 0.001669275 | 0.000391236 |
| Whisker Top (Max- Q3) | 0.00301252 | 0.002764737 | 0.001043297 | 0.00075639 |
| Whisker Bottom (Q1- Min) | 0.020722483 | 0.011711007 | 0.076017214 | 0.001499739 |

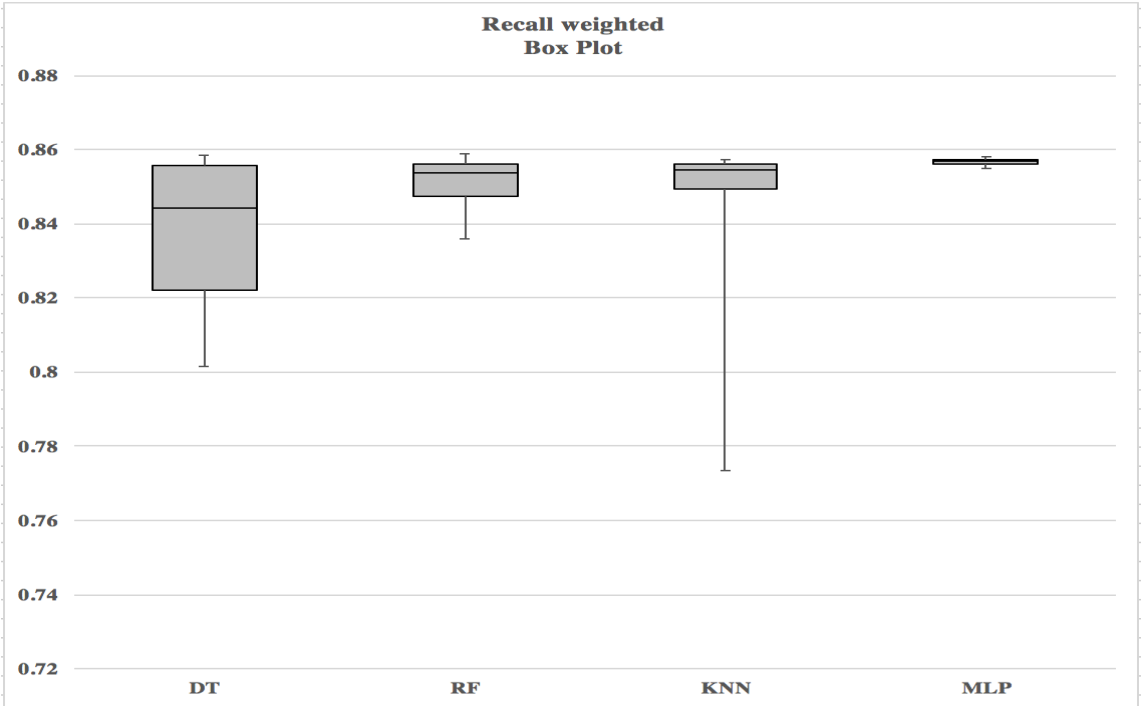


Figure 4.43. Recall Weight Box Plot for Heart Disease Dataset Using Suggested Features

#### 4.3.3.3 Best Model

The following diagram shows the best model of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithm using suggested features only for each evaluation metrics.

| Algorithm | Accuracy | | Precision Micro | | Precision Macro | | Precision Weighted | | Recall Micro | | Recall Macro | | Recall Weighted | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value |
| Decission Tree | max_depth: 4 | 0.858581116 | max_depth: 4 | 0.858581116 | max_depth: 4 | 0.837718605 | max_depth: 4 | 0.852828512 | max_depth: 4 | 0.858581116 | max_depth: 18 | 0.543696941 | max_depth: 4 | 0.858581116 |
| Random Forest | max_depth: 5 | 0.8590506 | max_depth: 5 | 0.8590506 | max_depth: 2 | 0.86990325 | max_depth: 2 | 0.859760048 | max_depth: 5 | 0.8590506 | max_depth: 20 | 0.548233019 | max_depth: 5 | 0.8590506 |
| KNN | n_neighbors: 14 | 0.85722483 | n_neighbors: 14 | 0.85722483 | n_neighbors: 20 | 0.748469484 | n_neighbors: 20 | 0.827417288 | n_neighbors: 14 | 0.85722483 | n_neighbors: 1 | 0.554452224 | n_neighbors: 14 | 0.85722483 |
| MLP | max_iter: 150000 | 0.857642149 | max_iter: 190000 | 0.857746479 | max_iter: 30000 | 0.816266373 | max_iter: 160000 | 0.844376754 | max_iter: 20000 | 0.857850809 | max_iter: 130000 | 0.52766627 | max_iter: 100000 | 0.857902973 |

Figure 4.44. Best Models for Heart Disease Dataset Using Suggested Features

#### 4.3.3.4 ROC Chart

Here we compared the ROC curve of the best models of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithm using suggested features only as follows:

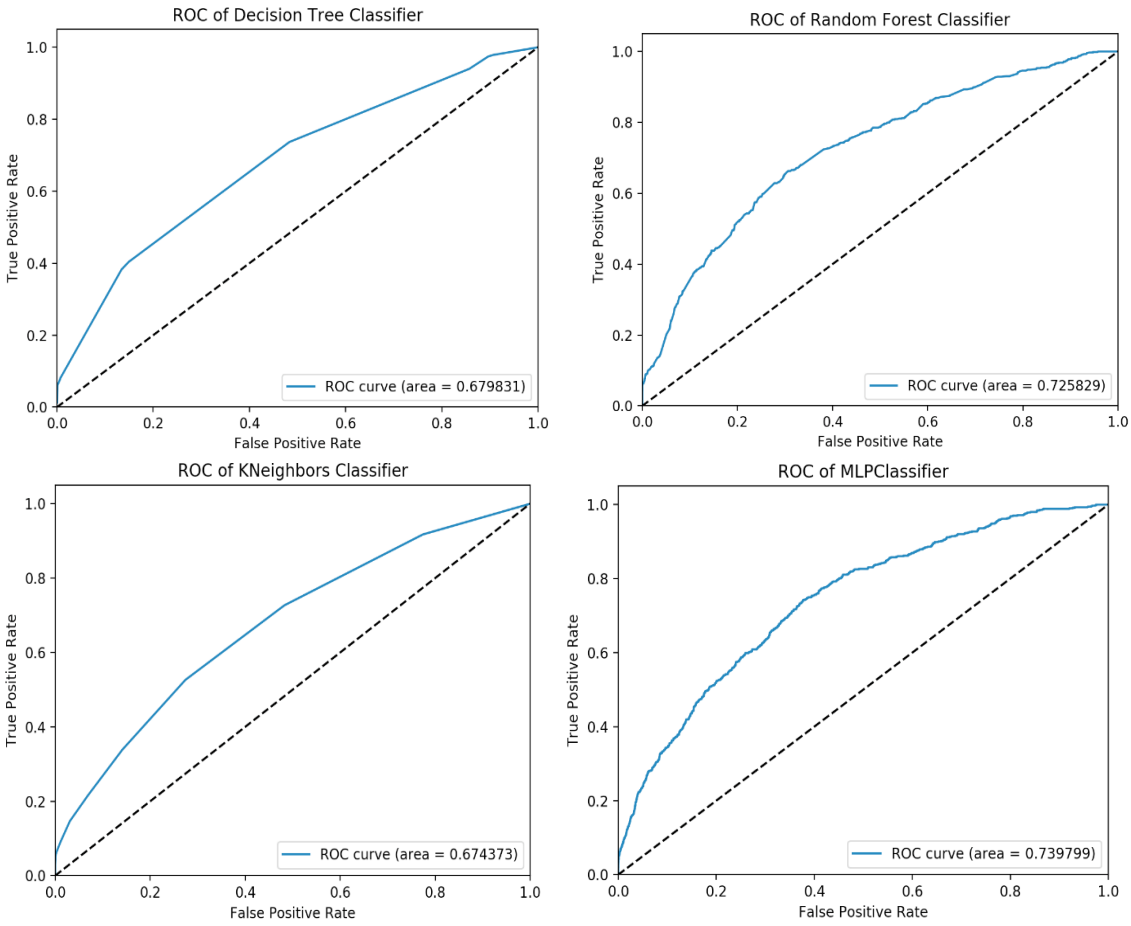


Figure 4.45. ROC Curve of Best Models for Heart Disease Dataset Using Suggested Features

### 4.3.4 Using Transfer Learning Combined with Suggested Features

For this experiment, we combined top 10 feature identified during transfer learning and experts suggested important features of heart disease dataset as shown in section 3.3. We used these combined features for training all the models of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithm and estimated the evaluation metrics for each model using grid search 5-fold cross-validation. Following sections shows the comparison between the models.

#### 4.3.4.1 Line Graph

Following line diagram shows the comparison of different models of each algorithm using combined features of transfer learning and expert suggested features.

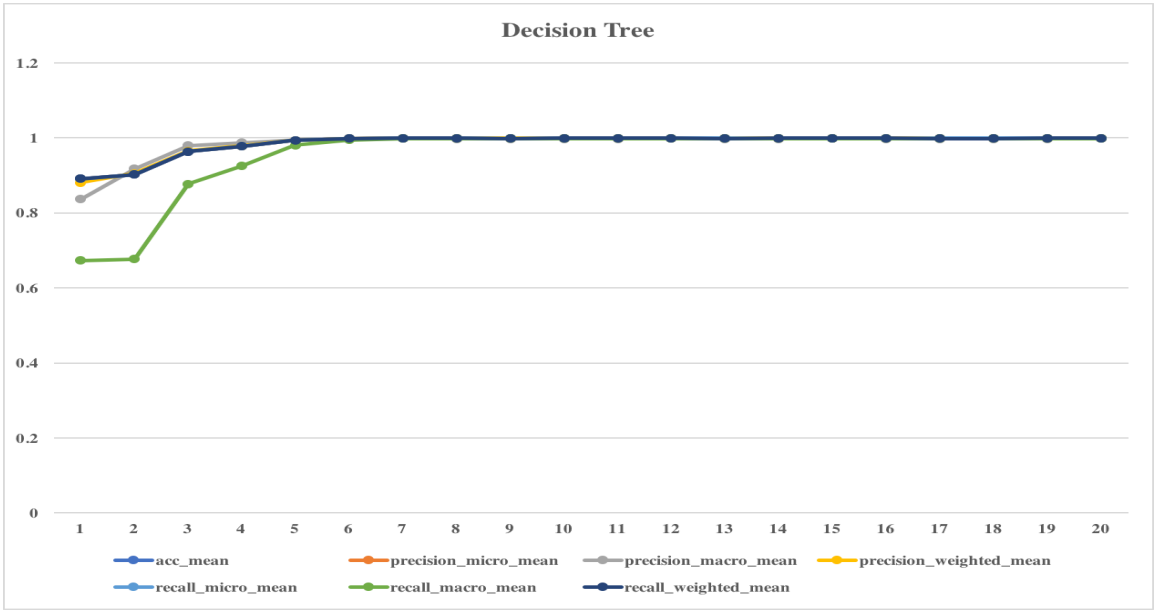


Figure 4.46. Line Graph of Decision Tree with Varying Max_Depth for Heart Disease Dataset Using Transfer Learning Combined with Suggested Features

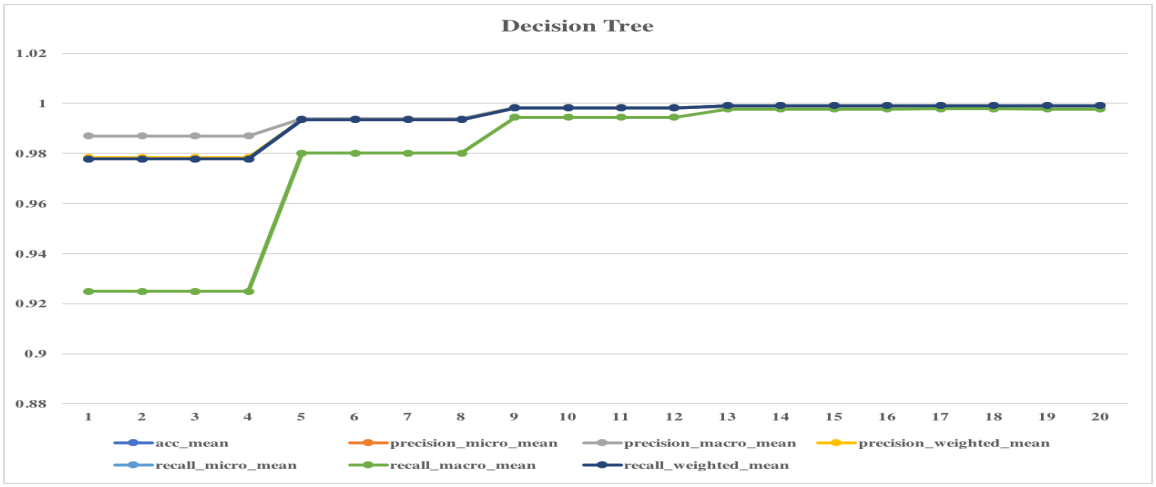


Figure 4.47. Line Graph of Decision Tree with Varying Max_Depth and Min_Sample_Split for Heart Disease Dataset Using Transfer Learning Combined with Suggested Features

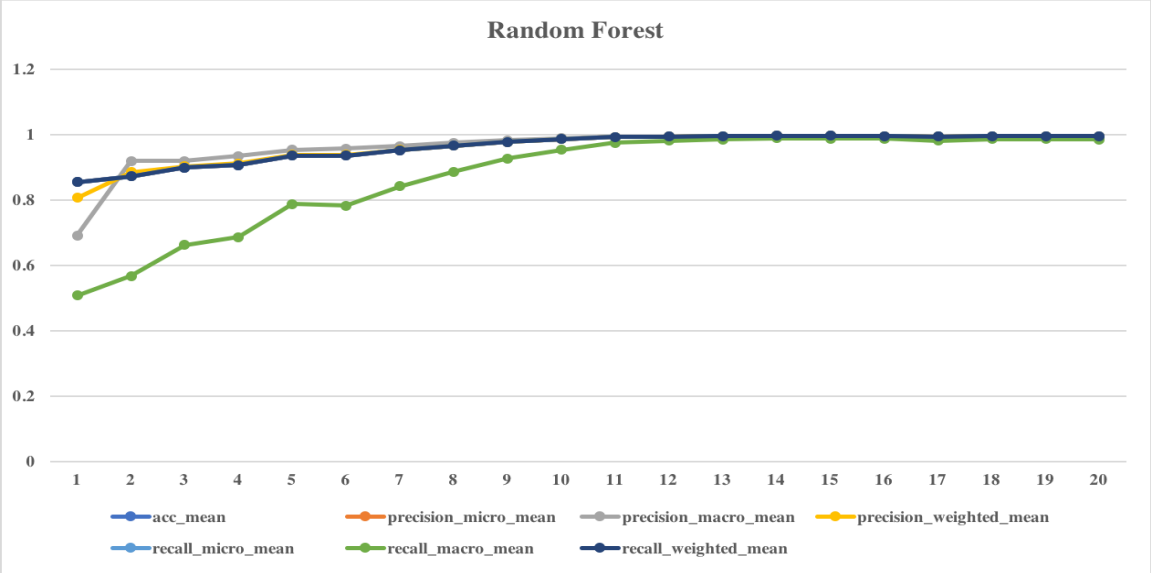


Figure 4.48. Line Graph of Random Forest with Varying Max_Depth for Heart Disease Dataset Using Transfer Learning Combined with Suggested Features

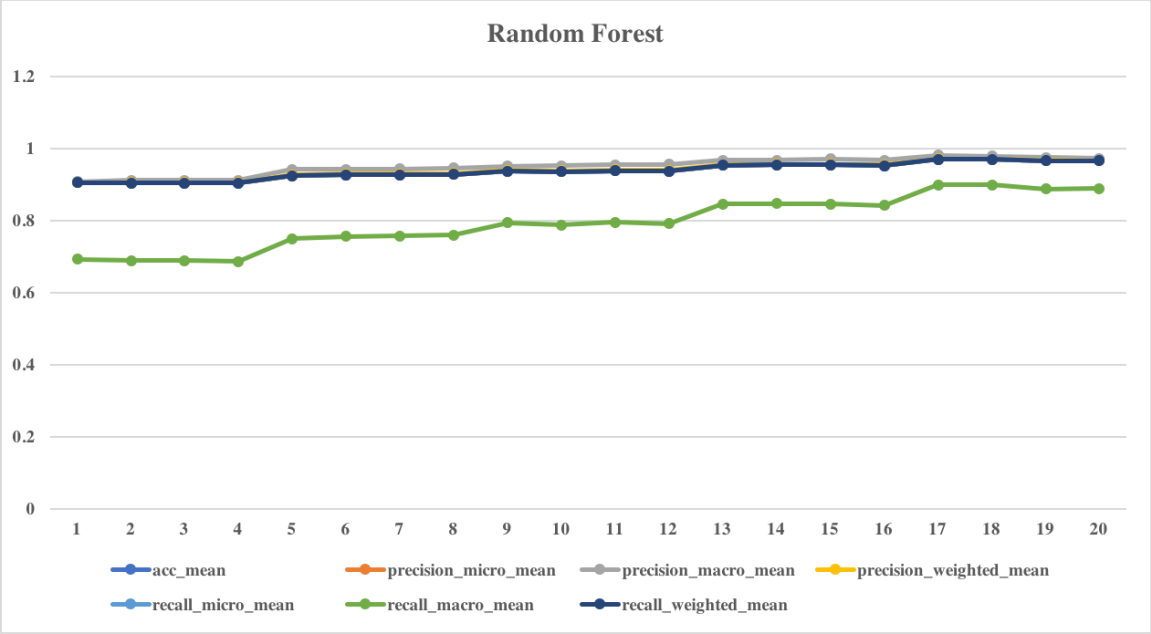


Figure 4.49. Line Graph of Random Forest with Varying Max_Depth and Min_Sample_Split for Heart Disease Dataset Using Transfer Learning Combined with Suggested Features

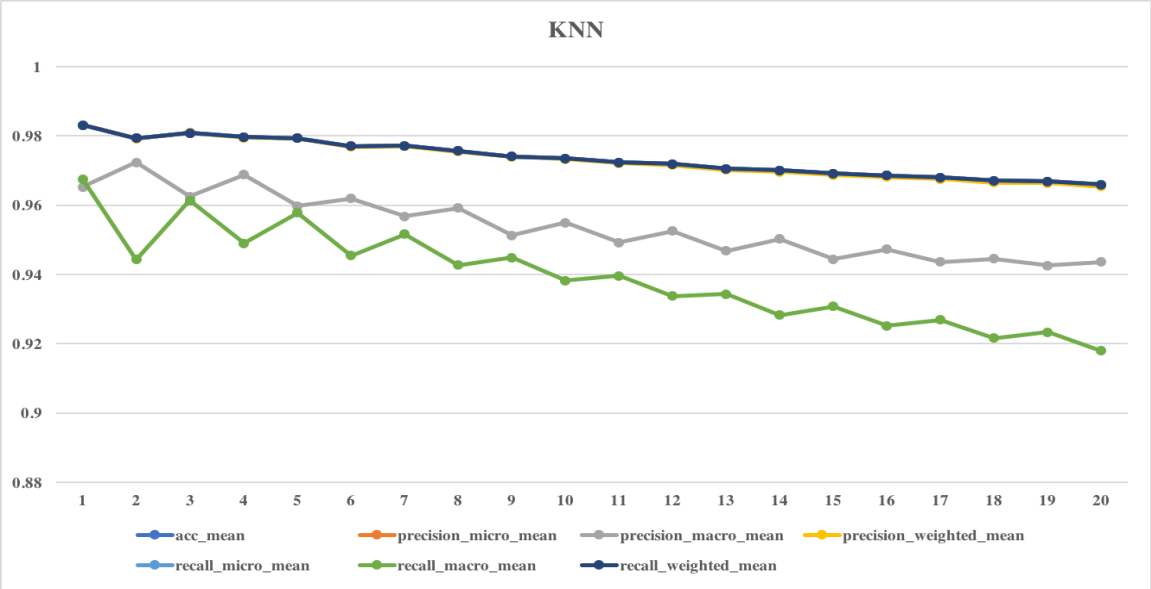


Figure 4.50. Line Graph of KNN with Varying N_Neighbor for Heart Disease Dataset Using Transfer Learning Combined with Suggested Features

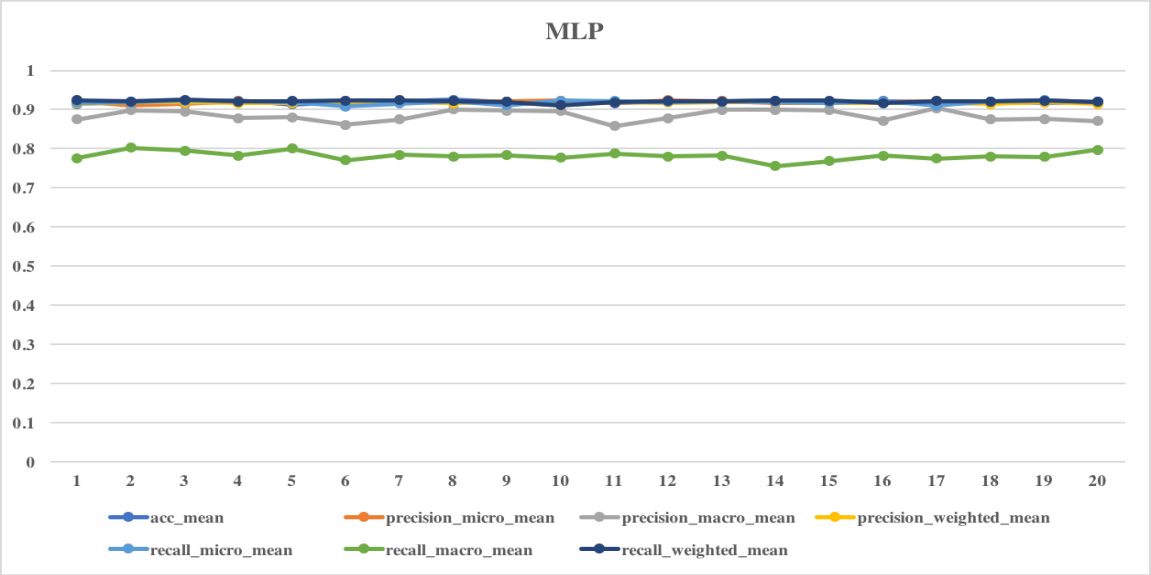


Figure 4.51. Line Graph of MLP with Varying Max_Iteration for Heart Disease Dataset Using Transfer Learning Combined with Suggested Features

#### 4.3.4.2 Box Plot

Following box plot shows the comparison of the evaluation metrics values for all the models created by varying max_depth of Decision Tree, max_depth of Random Forest, n_neighbor of K-Nearest Neighbor and max_iteration of MLP algorithms for heart disease dataset using combined features of transfer learning and expert suggested features.

Table 4.22. Accuracy Value for Heart Disease Dataset Using Transfer Learning And Expert Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.891079812 | 0.854929577 | 0.965988524 | 0.913354199 |
| First Quartile (Q1) | 0.996961398 | 0.935472092 | 0.968988002 | 0.918648931 |
| Median Value | 0.998904538 | 0.988706312 | 0.97295253 | 0.919640063 |
| Third Quartile(Q3) | 0.998969744 | 0.994600939 | 0.977699531 | 0.922026604 |
| Max Value | 0.999061033 | 0.995774648 | 0.983098592 | 0.924621805 |
| Box 1-hidden (Q1) | 0.996961398 | 0.935472092 | 0.968988002 | 0.918648931 |
| Box 2 (Median - Q1) | 0.00194314 | 0.05323422 | 0.003964528 | 0.000991132 |
| Box 3 (Q3-Median) | 6.52061E-05 | 0.005894627 | 0.004747001 | 0.002386541 |
| Whisker Top (Max- Q3) | 9.12885E-05 | 0.001173709 | 0.005399061 | 0.002595201 |
| Whisker Bottom (Q1- Min) | 0.105881586 | 0.080542514 | 0.002999478 | 0.005294731 |

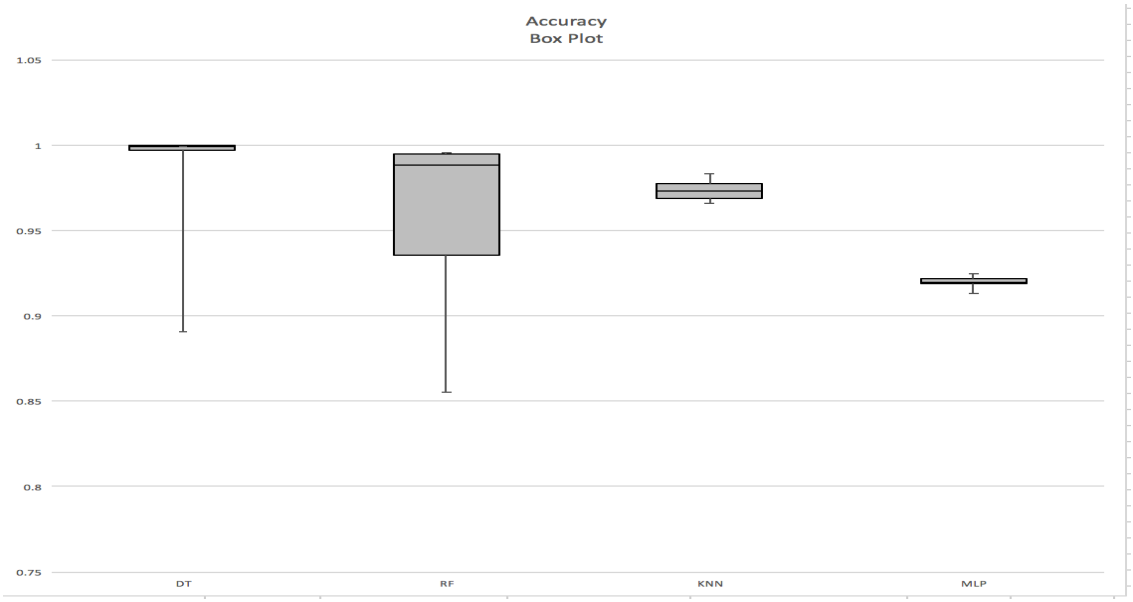


Figure 4.52. Accuracy Box Plot for Heart Disease Dataset Using Transfer Learning Combined with Suggested Features.

Table 4.23. Precision Macro Value for Heart Disease Dataset Using Transfer Learning And Expert Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.837102924 | 0.691384853 | 0.942542258 | 0.857540835 |
| First Quartile (Q1) | 0.99637893 | 0.955890129 | 0.946240199 | 0.874726392 |
| Median Value | 0.9976414 | 0.990777503 | 0.951952435 | 0.879196067 |
| Third Quartile(Q3) | 0.997853082 | 0.993249358 | 0.96031886 | 0.898390089 |
| Max Value | 0.998345457 | 0.995000378 | 0.972378469 | 0.903664631 |
| Box 1-hidden (Q1) | 0.99637893 | 0.955890129 | 0.946240199 | 0.874726392 |
| Box 2 (Median - Q1) | 0.00126247 | 0.034887374 | 0.005712237 | 0.004469675 |
| Box 3 (Q3- Median) | 0.000211682 | 0.002471855 | 0.008366425 | 0.019194023 |
| Whisker Top (Max- Q3) | 0.000492375 | 0.00175102 | 0.012059609 | 0.005274541 |
| Whisker Bottom (Q1- Min) | 0.159276006 | 0.264505276 | 0.003697941 | 0.017185557 |

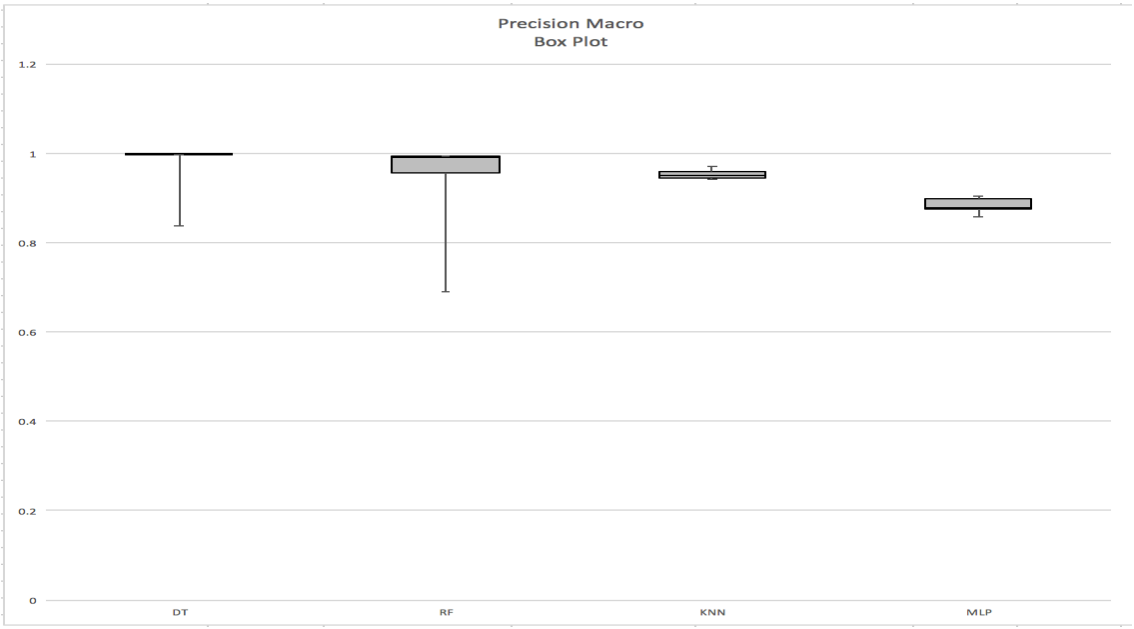


Figure 4.53. Precision Macro Box Plot for Heart Disease Dataset Using Transfer Learning Combined with Suggested Features

Table 4.24. Precision Micro Value for Heart Disease Dataset Using Transfer Learning And Expert Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.891079812 | 0.854929577 | 0.965988524 | 0.91152843 |
| First Quartile (Q1) | 0.997000522 | 0.935472092 | 0.968988002 | 0.918661972 |
| Median Value | 0.998956703 | 0.988706312 | 0.97295253 | 0.919874804 |
| Third Quartile(Q3) | 0.999021909 | 0.994600939 | 0.977699531 | 0.922157016 |
| Max Value | 0.999113198 | 0.995774648 | 0.983098592 | 0.923943662 |
| Box 1-hidden (Q1) | 0.997000522 | 0.935472092 | 0.968988002 | 0.918661972 |
| Box 2 (Median - Q1) | 0.001956182 | 0.05323422 | 0.003964528 | 0.001212833 |
| Box 3 (Q3- Median) | 6.52061E-05 | 0.005894627 | 0.004747001 | 0.002282212 |
| Whisker Top (Max- Q3) | 9.12885E-05 | 0.001173709 | 0.005399061 | 0.001786646 |
| Whisker Bottom (Q1- Min) | 0.105920709 | 0.080542514 | 0.002999478 | 0.007133542 |

.

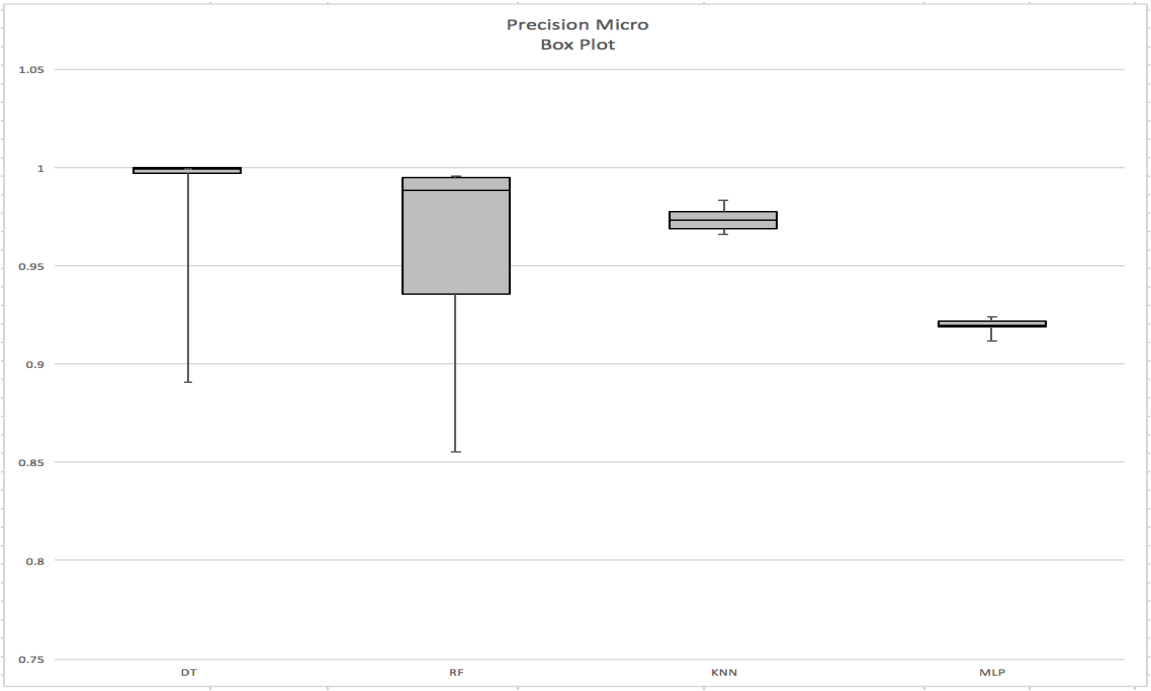


Figure 4.54. Precision Micro Box Plot for Heart Disease Dataset Using Transfer Learning Combined with Suggested Features

Table 4.25. Precision Weighted Value for Heart Disease Dataset Using Transfer Learning And Expert Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.881309714 | 0.80677865 | 0.965387758 | 0.911269149 |
| First Quartile (Q1) | 0.996979189 | 0.938320385 | 0.968600456 | 0.916106893 |
| Median Value | 0.998909583 | 0.988797921 | 0.972722166 | 0.917398043 |
| Third Quartile(Q3) | 0.999011643 | 0.994590823 | 0.977603181 | 0.918733287 |
| Max Value | 0.999168132 | 0.995768866 | 0.983179918 | 0.920105396 |
| Box 1-hidden (Q1) | 0.996979189 | 0.938320385 | 0.968600456 | 0.916106893 |
| Box 2 (Median - Q1) | 0.001930393 | 0.050477535 | 0.00412171 | 0.00129115 |
| Box 3 (Q3- Median) | 0.000102061 | 0.005792902 | 0.004881015 | 0.001335244 |
| Whisker Top (Max- Q3) | 0.000156488 | 0.001178043 | 0.005576737 | 0.001372109 |
| Whisker Bottom (Q1- Min) | 0.115669475 | 0.131541735 | 0.003212698 | 0.004837744 |

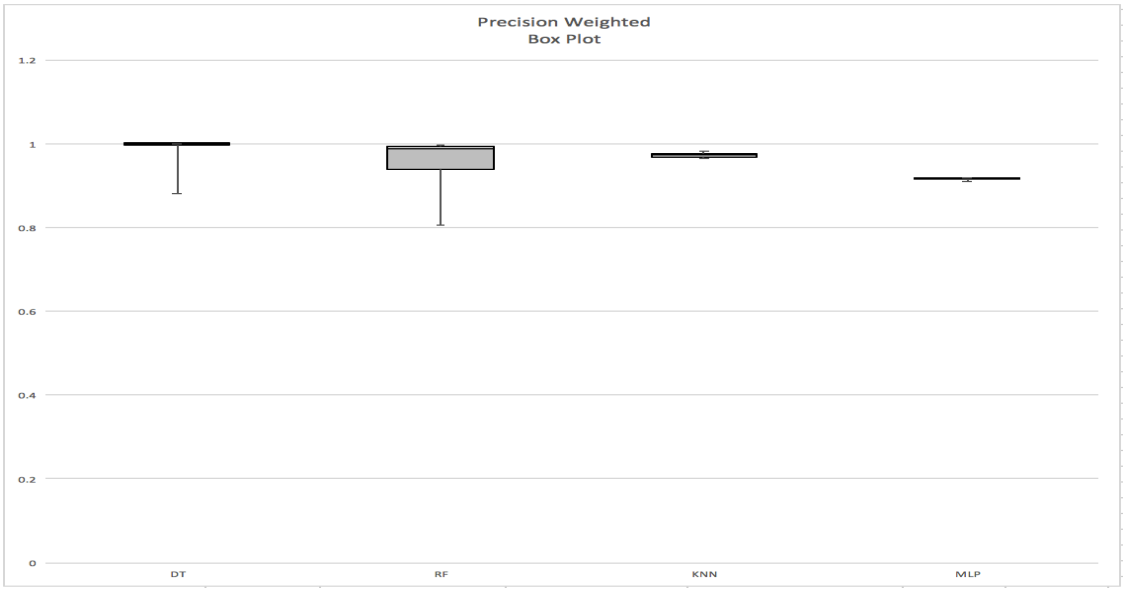


Figure 4.55. Precision Weighted Box Plot for Heart Disease Dataset Using Transfer Learning Combined with Suggested Features

Table 4.26. Recall Macro Value for Heart Disease Dataset Using Transfer Learning and Expert Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.673276336 | 0.507449099 | 0.917938033 | 0.756196078 |
| First Quartile (Q1) | 0.990946322 | 0.786000781 | 0.927917871 | 0.776304338 |
| Median Value | 0.998167541 | 0.964082678 | 0.938907164 | 0.780978741 |
| Third Quartile(Q3) | 0.998243997 | 0.985190309 | 0.946319003 | 0.785042891 |
| Max Value | 0.998305143 | 0.988859848 | 0.967623898 | 0.802112184 |
| Box 1-hidden (Q1) | 0.990946322 | 0.786000781 | 0.927917871 | 0.776304338 |
| Box 2 (Median - Q1) | 0.007221219 | 0.178081897 | 0.010989293 | 0.004674403 |
| Box 3 (Q3- Median) | 7.64564E-05 | 0.021107632 | 0.007411839 | 0.004064151 |
| Whisker Top (Max- Q3) | 6.11461E-05 | 0.003669538 | 0.021304895 | 0.017069293 |
| Whisker Bottom (Q1- Min) | 0.317669986 | 0.278551682 | 0.009979838 | 0.02010826 |

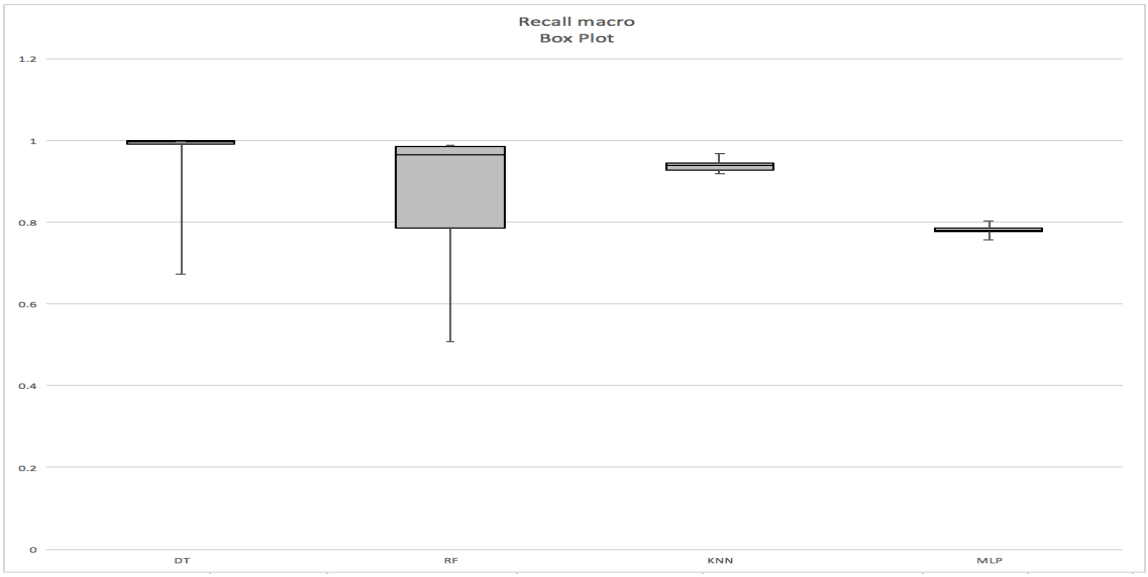


Figure 4.56. Recall Macro Plot for Heart Disease Dataset Using Transfer Learning Combined with Suggested Features

Table 4.27. Recall Micro Value for Heart Disease Dataset Using Transfer Learning and Expert Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.891079812 | 0.854929577 | 0.965988524 | 0.90808555 |
| First Quartile (Q1) | 0.997000522 | 0.935472092 | 0.968988002 | 0.917957746 |
| Median Value | 0.998878456 | 0.988706312 | 0.97295253 | 0.919848722 |
| Third Quartile(Q3) | 0.999008868 | 0.994600939 | 0.977699531 | 0.921244131 |
| Max Value | 0.999061033 | 0.995774648 | 0.983098592 | 0.923735003 |
| Box 1-hidden (Q1) | 0.997000522 | 0.935472092 | 0.968988002 | 0.917957746 |
| Box 2 (Median - Q1) | 0.001877934 | 0.05323422 | 0.003964528 | 0.001890975 |
| Box 3 (Q3- Median) | 0.000130412 | 0.005894627 | 0.004747001 | 0.001395409 |
| Whisker Top (Max- Q3) | 5.21648E-05 | 0.001173709 | 0.005399061 | 0.002490871 |
| Whisker Bottom (Q1- Min) | 0.105920709 | 0.080542514 | 0.002999478 | 0.009872196 |

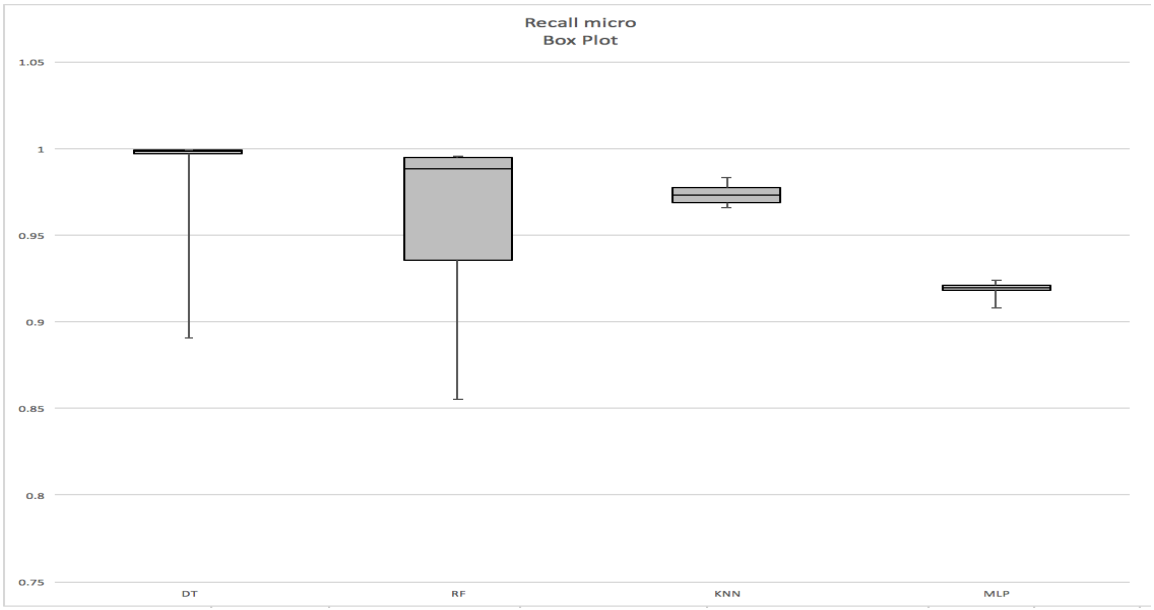


Figure 4.57. Recall Micro Box Plot for Heart Disease Dataset Using Transfer Learning Combined with Suggested Features

Table 4.28. Recall Weighted Value for Heart Disease Dataset Using Transfer Learning and Expert Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.891079812 | 0.854929577 | 0.965988524 | 0.910745957 |
| First Quartile (Q1) | 0.996922274 | 0.935472092 | 0.968988002 | 0.920357329 |
| Median Value | 0.998878456 | 0.988706312 | 0.97295253 | 0.921961398 |
| Third Quartile(Q3) | 0.999008868 | 0.994600939 | 0.977699531 | 0.922613459 |
| Max Value | 0.999165363 | 0.995774648 | 0.983098592 | 0.925299948 |
| Box 1-hidden (Q1) | 0.996922274 | 0.935472092 | 0.968988002 | 0.920357329 |
| Box 2 (Median - Q1) | 0.001956182 | 0.05323422 | 0.003964528 | 0.001604069 |
| Box 3 (Q3-Median) | 0.000130412 | 0.005894627 | 0.004747001 | 0.000652061 |
| Whisker Top (Max- Q3) | 0.000156495 | 0.001173709 | 0.005399061 | 0.002686489 |
| Whisker Bottom (Q1- Min) | 0.105842462 | 0.080542514 | 0.002999478 | 0.009611372 |

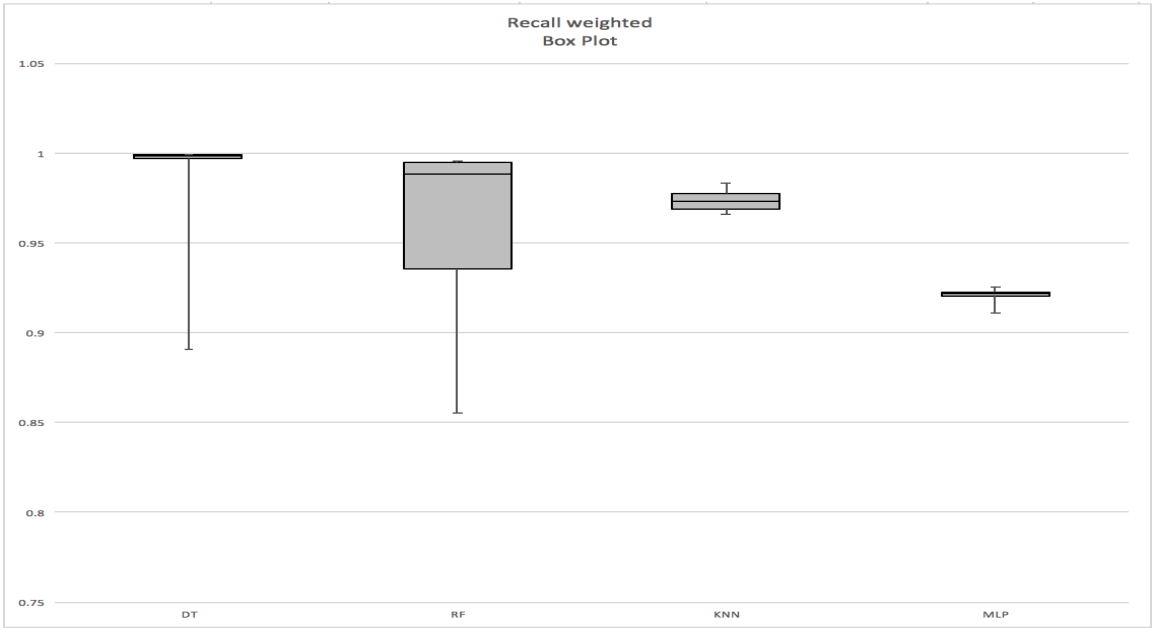


Figure 4.58. Recall Weighted Box Plot for Heart Disease Dataset Using Transfer Learning Combined with Suggested Features

#### 4.3.4.3 Best Model

The following diagram shows the best model of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithm using combined features of transfer learning and expert suggested features for each evaluation metrics.

| Algorithm | Accuracy | | Precision Micro | | Precision Macro | | Precision Weighted | | Recall Micro | | Recall Macro | | Recall Weighted | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value |
| Decission Tree | max_depth: 8 | 0.999061033 | max_depth: 11 | 0.999113198 | max_depth: 20 | 0.998345457 | max_depth: 15 | 0.999168132 | max_depth: 12 | 0.999061033 | max_depth: 12 | 0.998305143 | max_depth: 12 | 0.999165363 |
| Random Forest | max_depth: 14 | 0.995774648 | max_depth: 14 | 0.995774648 | max_depth: 15 | 0.995000378 | max_depth: 14 | 0.995768866 | max_depth: 14 | 0.995774648 | max_depth: 14 | 0.988859848 | max_depth: 14 | 0.995774648 |
| KNN | n_neighbors: 1 | 0.983098592 | n_neighbors: 1 | 0.983098592 | n_neighbors: 2 | 0.972378469 | n_neighbors: 1 | 0.983179918 | n_neighbors: 1 | 0.983098592 | n_neighbors: 1 | 0.967623898 | n_neighbors: 1 | 0.983098592 |
| MLP | max_iter: 80000 | 0.924621805 | max_iter: 120000 | 0.923943662 | max_iter: 17000 | 0.903664631 | max_iter: 110000 | 0.920105396 | max_iter: 190000 | 0.923735003 | max_iter: 20000 | 0.802112184 | max_iter: 30000 | 0.925299948 |

Figure 4.59. Best Models for Heart Disease Dataset Using Transfer Learning Combined with Suggested Features

#### 4.3.4.4 ROC Chart

Here we compared the ROC curve of the best models of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithm using combined features of transfer learning and expert suggested features as follows:

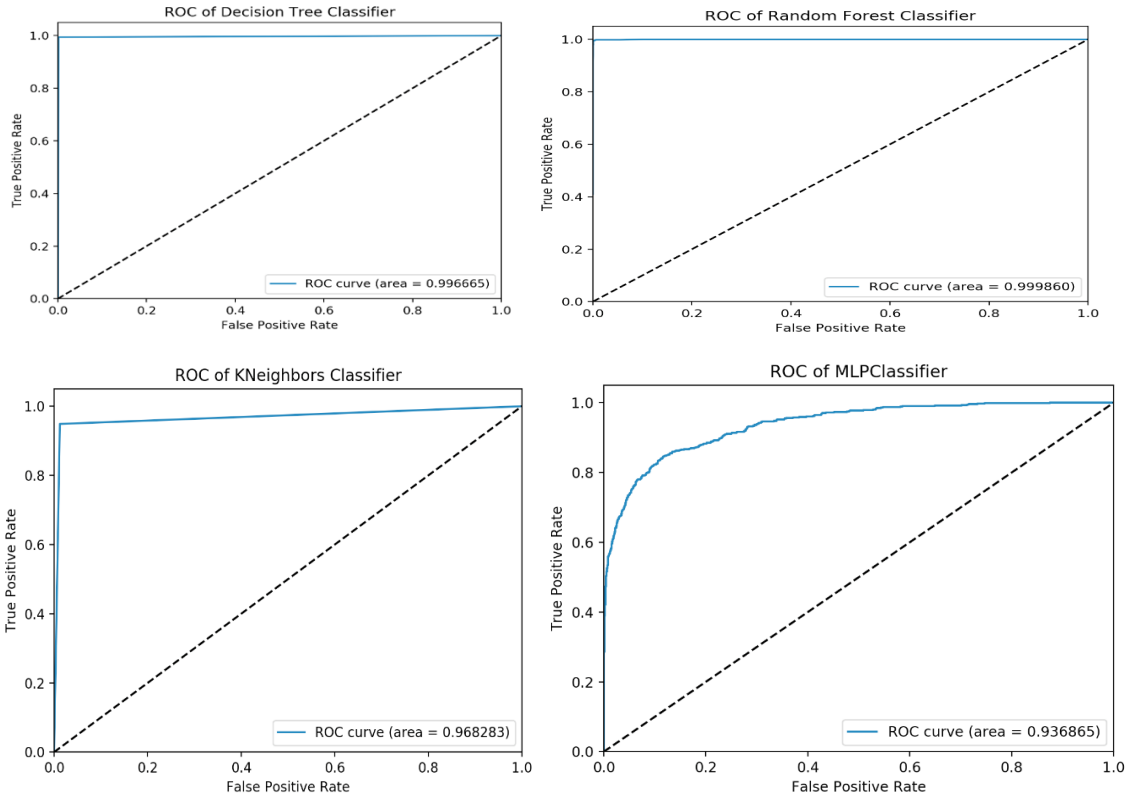


Figure 4.60. ROC Curve of Best Model for Heart Disease Dataset Using Transfer Learning Combined with Suggested Features

### 4.3.5 Methodology and Algorithm Comparison Based on Accuracy for Heart Disease Dataset

In this section, we compare all the 4-methodologies used with heart disease dataset, following table shows the comparisons between best models created by modifying one coefficient for each methodology and each machine learning algorithms.

Table 4.29. Accuracy Based Comparisons of Best Model for each Methodology and each Machine Learning Algorithms

| Features Used for Training | Best Model Accuracy with Grid Search Evaluation | | | |
|---|---|---|---|---|
| | Decision Tree | Random Forest | KNN | MLP |
| **All 53 Features** | 0.998435054773 | 0.971465832029 | 0.952164840897 | 0.922222222222 |
| **Transfer Learning with Top 10 Features** | 0.999113197705 | 0.998539384455 | 0.982994261868 | 0.921178925404 |
| **With Expert Suggested Features** | 0.858581116328 | 0.859050599896 | 0.857224830464 | 0.857642149191 |
| **Transfer Learning with Top 10 and suggested Features** | 0.999061032864 | 0.995774647887 | 0.983098591549 | 0.924621804903 |

The table 4 shows, the transfer learning methodology has better or almost same accuracy for all the machine learning algorithms comparing to other methodology. Here, it also shows that the Decision Tree algorithm outperformed to be the best among all the machine learning algorithms for all the methodology. It also shows that the model trained with only expert suggested features has the lowest accuracy for each algorithm. It also concludes that if the suggested features are combined with transfer learning, it outperformed to be the best among all the methodology.

### 4.4 Readmission Dataset

We did different experiments on the readmission dataset by considering the machine learning algorithms and transfer learning technique. The description of the readmission dataset is explained at appendix B.

### 4.5 Result and Finding with Readmission Dataset

In this section, we mention detail results obtained for readmission dataset by following methodologies specified in chapter 3.

#### 4.5.1 Using All the Features of Dataset

In this technique, we created the different models of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithms using all the feature of the dataset. And each trained model

performance is estimated by calculation evaluation metrics using grid search with 5-fold cross validation. We also compared the different models of the different algorithm as shown below.

#### 4.5.1.1 Line Graph

Following line diagram shows the comparison of different models of each algorithm based on evaluation metrics.

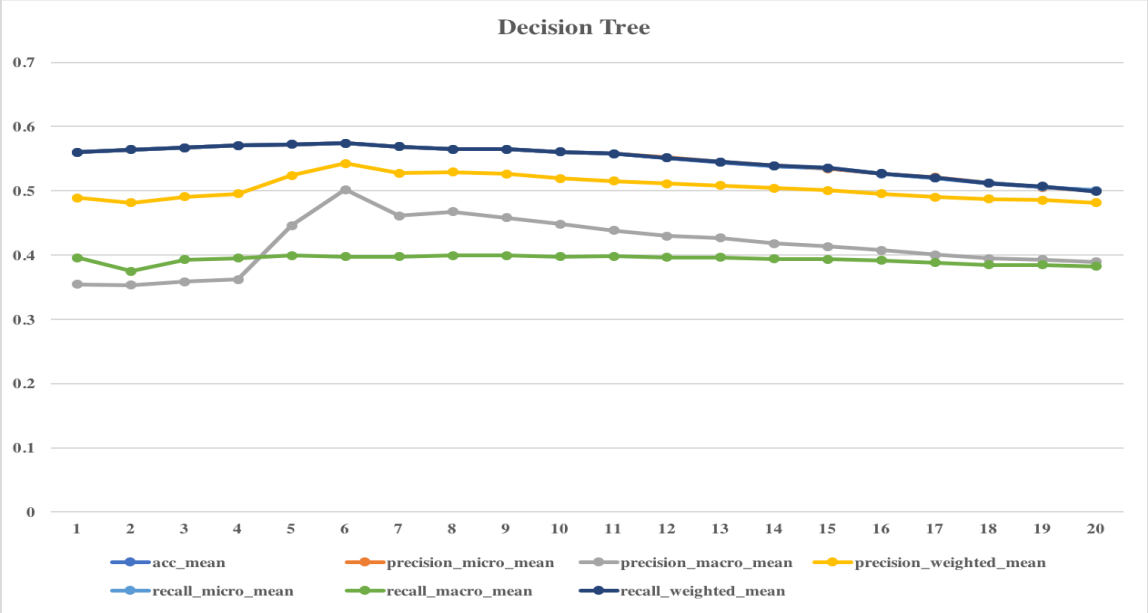


Figure 4.61. Line Graph of Decision Tree with Varying Max_Depth for Readmission Dataset Using All Features

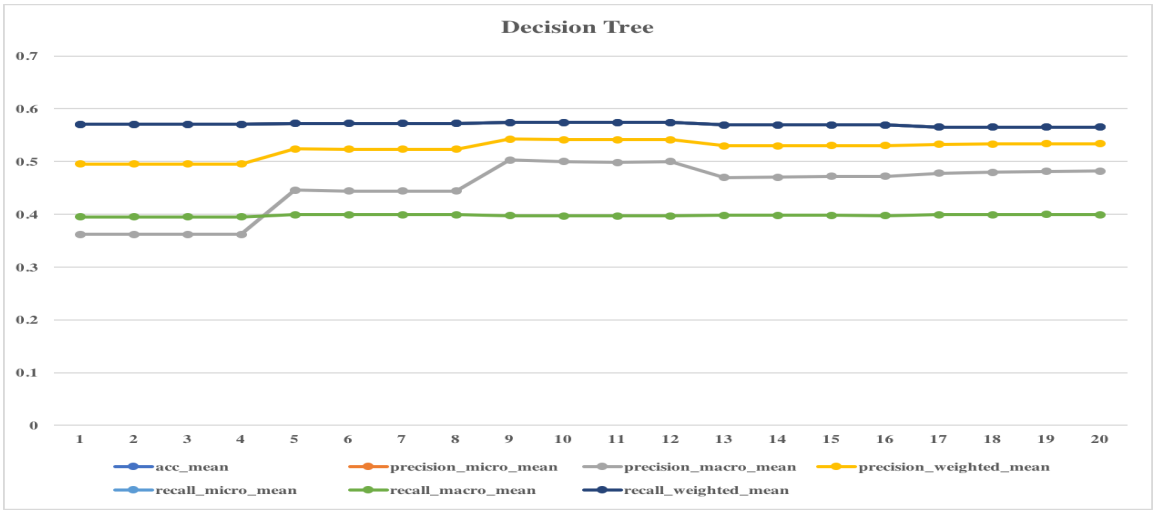


Figure 4.62. Line Graph of Decision Tree with Varying Max_Depth and Min_Sample_Split for Readmission Dataset Using All Features

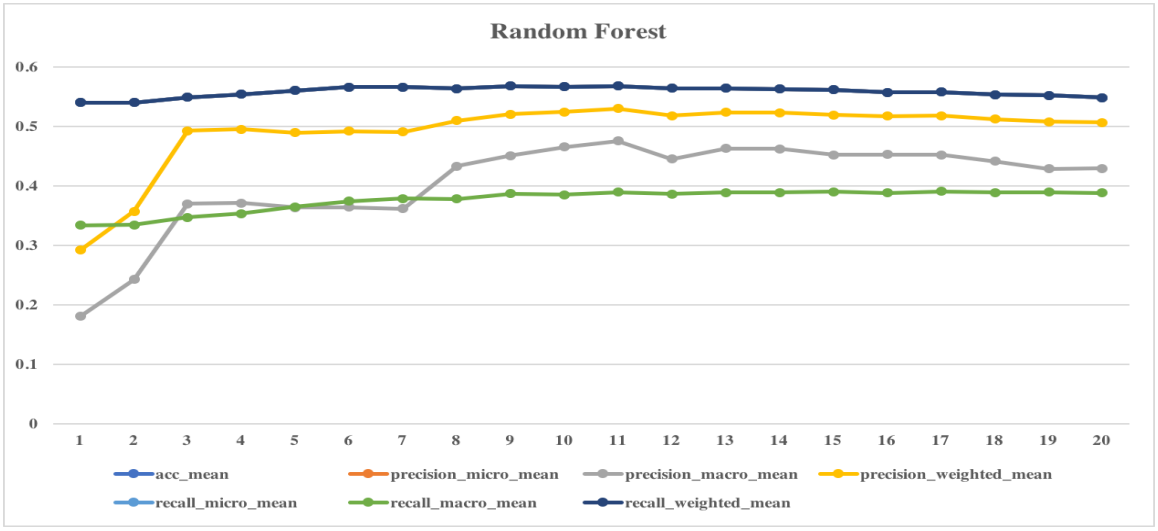


Figure 4.63. Line Graph of Random Forest with Varying Max_Depth for Readmission Dataset Using All Features.

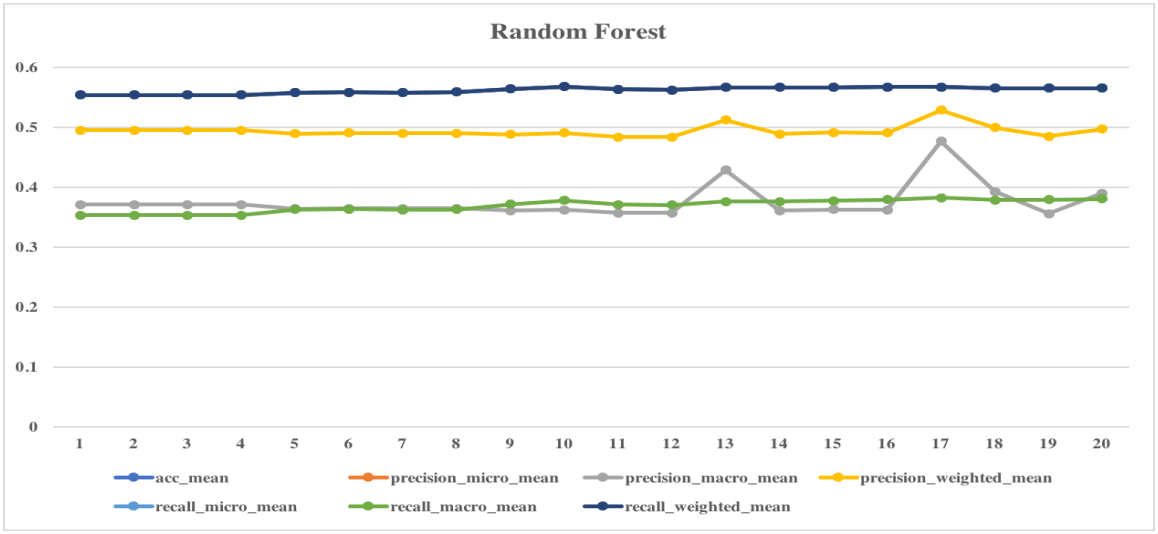


Figure 4.64. Line Graph of Random Forest with Varying Max_Depth and Min_Sample_Split for Readmission Dataset Using All Features

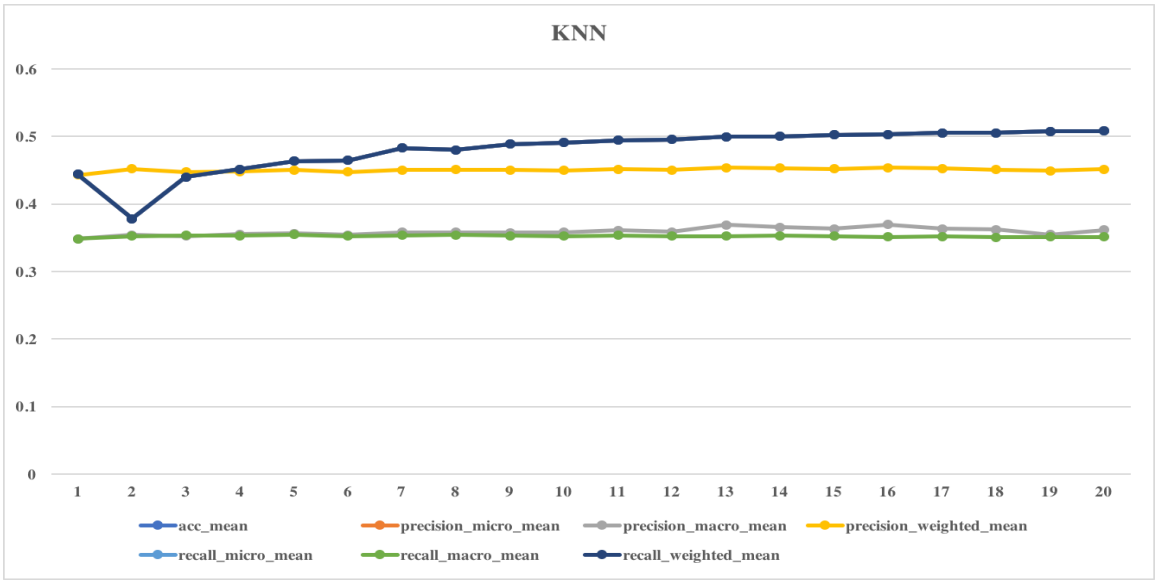


Figure 4.65. Line Graph of KNN with Varying N_Neighbor for Readmission Dataset Using All Features

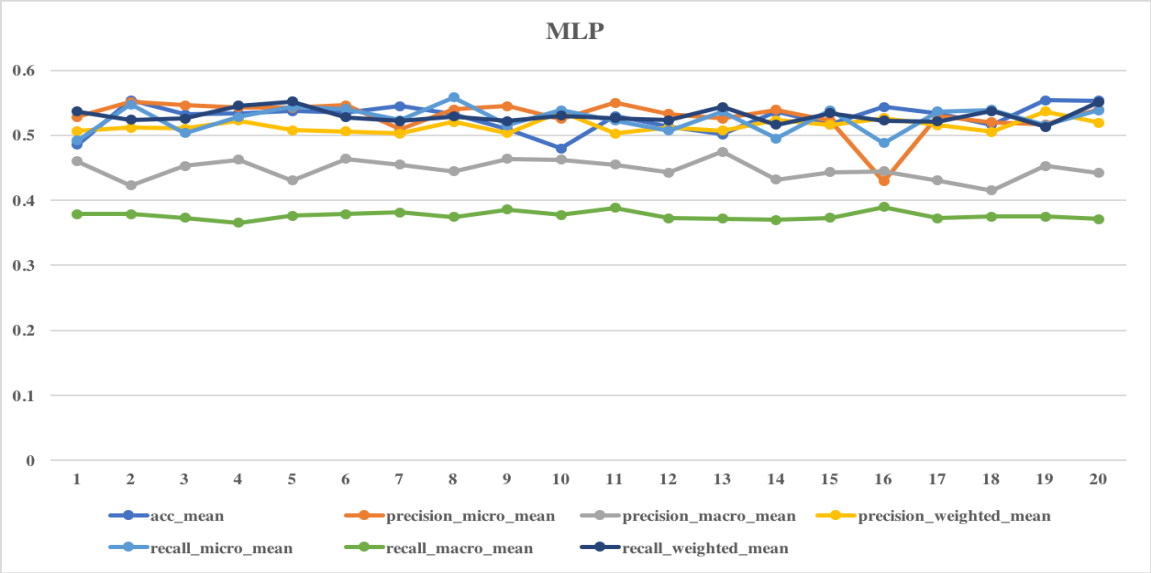


Figure 4.66. Line Graph of MLP with Varying Max_Iteration for Readmission Dataset Using All Features

#### 4.5.1.2 Box Plot

Following box plot shows the comparison of the evaluation metrics values for all the models created by varying max_depth of Decision Tree, max_depth of Random Forest, n_neighbor of K-Nearest Neighbor and max_iteration of MLP algorithms for readmission dataset using all features.

Table 4.30. Accuracy Value for Readmission Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.500384627 | 0.539786407 | 0.378462891 | 0.479654735 |
| First Quartile (Q1) | 0.532670809 | 0.552688892 | 0.464574363 | 0.514967981 |
| Median Value | 0.558907859 | 0.560791033 | 0.492831953 | 0.532853132 |
| Third Quartile(Q3) | 0.565306653 | 0.564397534 | 0.502542534 | 0.538425228 |
| Max Value | 0.573843372 | 0.56797906 | 0.508436816 | 0.553912705 |
| Box 1-hidden (Q1) | 0.532670809 | 0.552688892 | 0.464574363 | 0.514967981 |
| Box 2 (Median - Q1) | 0.02623705 | 0.008102141 | 0.02825759 | 0.017885151 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Box 3 (Q3-Median) | 0.006398793 | 0.003606502 | 0.009710581 | 0.005572095 |
| Whisker Top (Max- Q3) | 0.008536719 | 0.003581526 | 0.005894283 | 0.015487477 |
| Whisker Bottom (Q1- Min) | 0.032286182 | 0.012902485 | 0.086111472 | 0.035313246 |

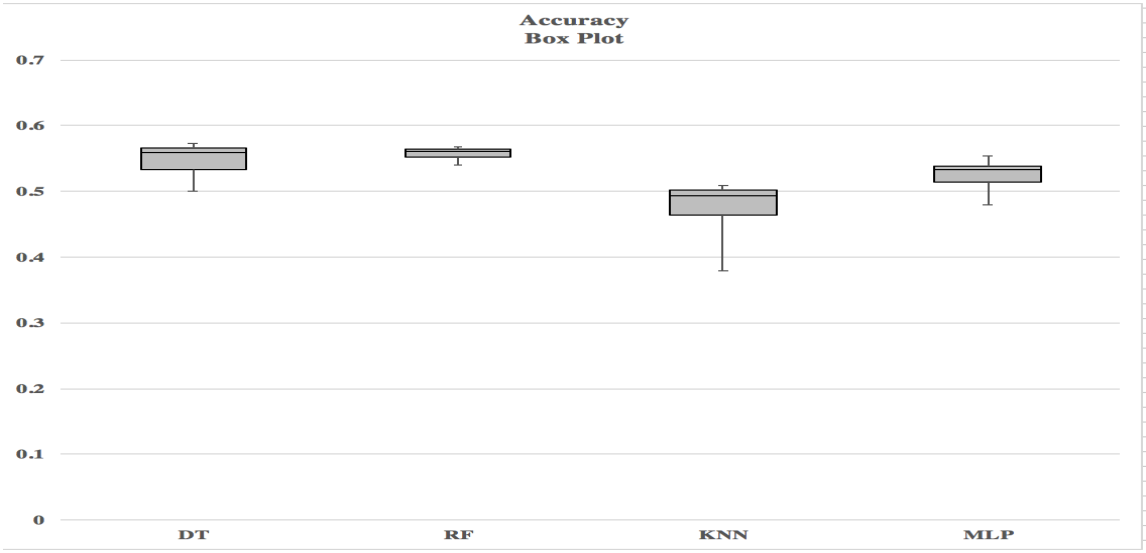


Figure 4.67. Accuracy Box Plot for Readmission Dataset Using All Features

Table 4.31. Precision Macro Value for Readmission Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.353411859 | 0.179928802 | 0.348460019 | 0.415094365 |
| First Quartile (Q1) | 0.391724824 | 0.368185748 | 0.355145753 | 0.439412503 |
| Median Value | 0.41539977 | 0.436940101 | 0.358454531 | 0.44875765 |
| Third Quartile(Q3) | 0.446147512 | 0.451984185 | 0.362344689 | 0.460352895 |
| Max Value | 0.501806474 | 0.475013144 | 0.369646216 | 0.474367028 |
| Box 1-hidden (Q1) | 0.391724824 | 0.368185748 | 0.355145753 | 0.439412503 |
| Box 2 (Median - Q1) | 0.023674946 | 0.068754352 | 0.003308778 | 0.009345147 |
| Box 3 (Q3-Median) | 0.030747741 | 0.015044084 | 0.003890158 | 0.011595245 |
| Whisker Top (Max- Q3) | 0.055658963 | 0.02302896 | 0.007301528 | 0.014014133 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Whisker Bottom (Q1- Min) | 0.038312965 | 0.188256946 | 0.006685734 | 0.024318137 |

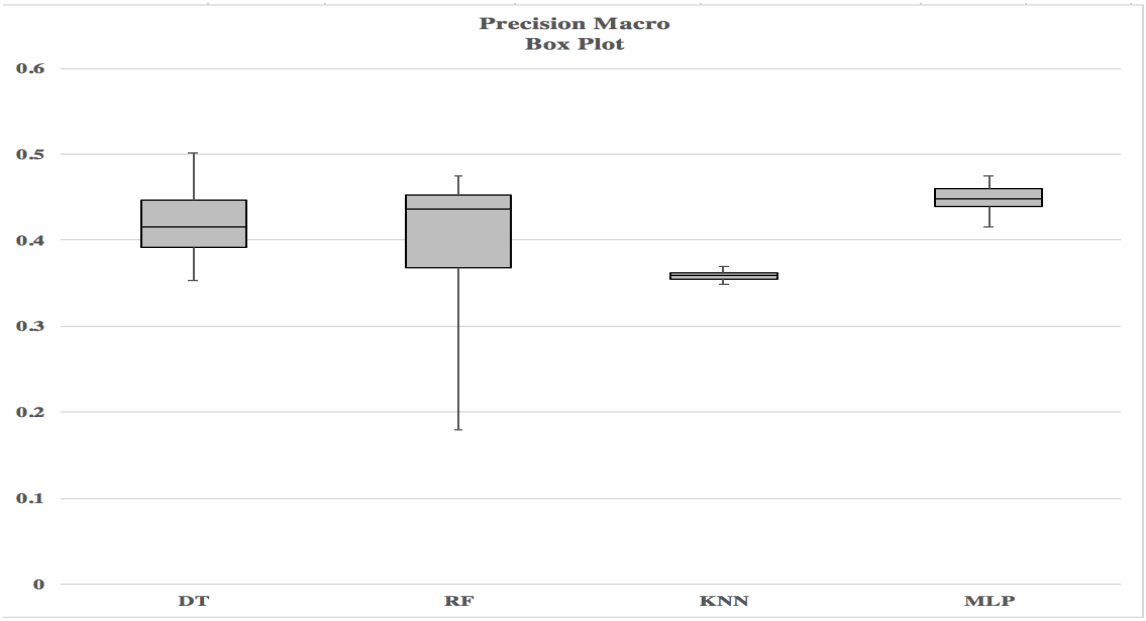


Figure 4.68. Precision Macro Box Plot for Readmission Dataset Using All Features

Table 4.32. Precision Micro Value for Readmission Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.499785208 | 0.539786407 | 0.378462891 | 0.429353527 |
| First Quartile (Q1) | 0.532256211 | 0.552688892 | 0.464574363 | 0.524278949 |
| Median Value | 0.558922845 | 0.560791033 | 0.492831953 | 0.535675395 |
| Third Quartile(Q3) | 0.565246711 | 0.564397534 | 0.502542534 | 0.543387914 |
| Max Value | 0.573853362 | 0.56797906 | 0.508436816 | 0.550915612 |
| Box 1-hidden (Q1) | 0.532256211 | 0.552688892 | 0.464574363 | 0.524278949 |
| Box 2 (Median - Q1) | 0.026666633 | 0.008102141 | 0.02825759 | 0.011396445 |
| Box 3 (Q3- Median) | 0.006323866 | 0.003606502 | 0.009710581 | 0.007712519 |
| Whisker Top (Max- Q3) | 0.008606652 | 0.003581526 | 0.005894283 | 0.007527698 |
| Whisker Bottom (Q1- Min) | 0.032471003 | 0.012902485 | 0.086111472 | 0.094925422 |

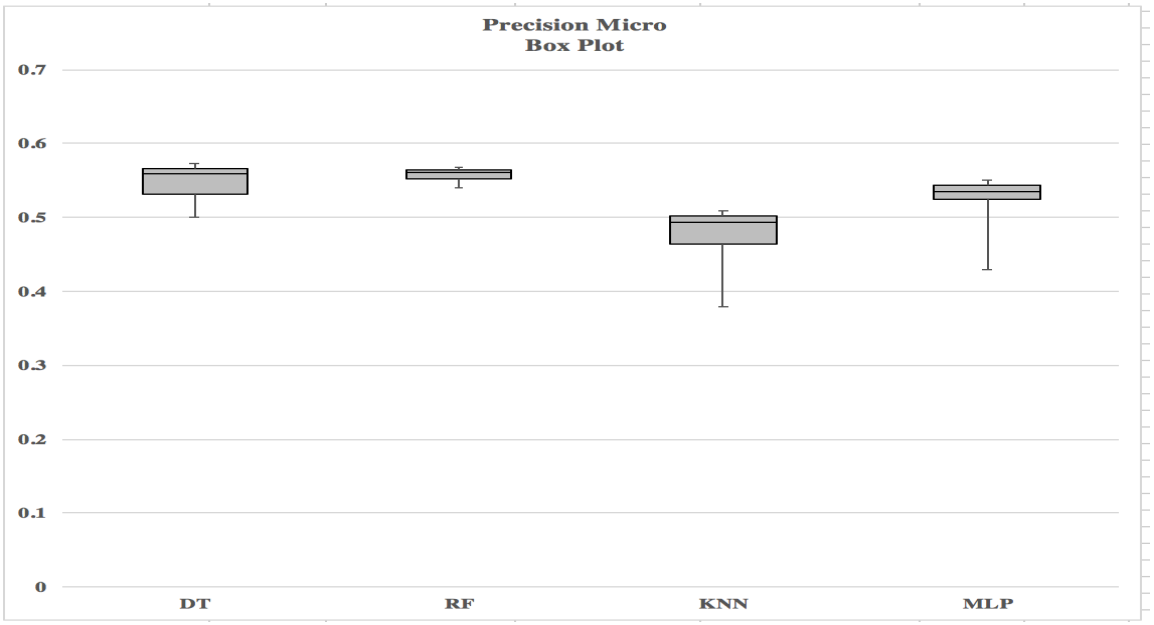


Figure 4.69. Precision Micro Box Plot for Readmission Dataset Using All Features

Table 4.33. Precision Weighted Value for Readmission Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.481312113 | 0.291369365 | 0.443030809 | 0.502406752 |
| First Quartile (Q1) | 0.489924537 | 0.492246838 | 0.449759361 | 0.506339407 |
| Median Value | 0.502199878 | 0.510643638 | 0.450790012 | 0.511753291 |
| Third Quartile(Q3) | 0.520266294 | 0.519463763 | 0.452243777 | 0.520864211 |
| Max Value | 0.542467919 | 0.529759957 | 0.453840202 | 0.536597265 |
| Box 1-hidden (Q1) | 0.489924537 | 0.492246838 | 0.449759361 | 0.506339407 |
| Box 2 (Median - Q1) | 0.01227534 | 0.0183968 | 0.00103065 | 0.005413884 |
| Box 3 (Q3- Median) | 0.018066416 | 0.008820125 | 0.001453765 | 0.00911092 |
| Whisker Top (Max- Q3) | 0.022201625 | 0.010296194 | 0.001596426 | 0.015733055 |
| Whisker Bottom (Q1- Min) | 0.008612424 | 0.200877472 | 0.006728552 | 0.003932655 |

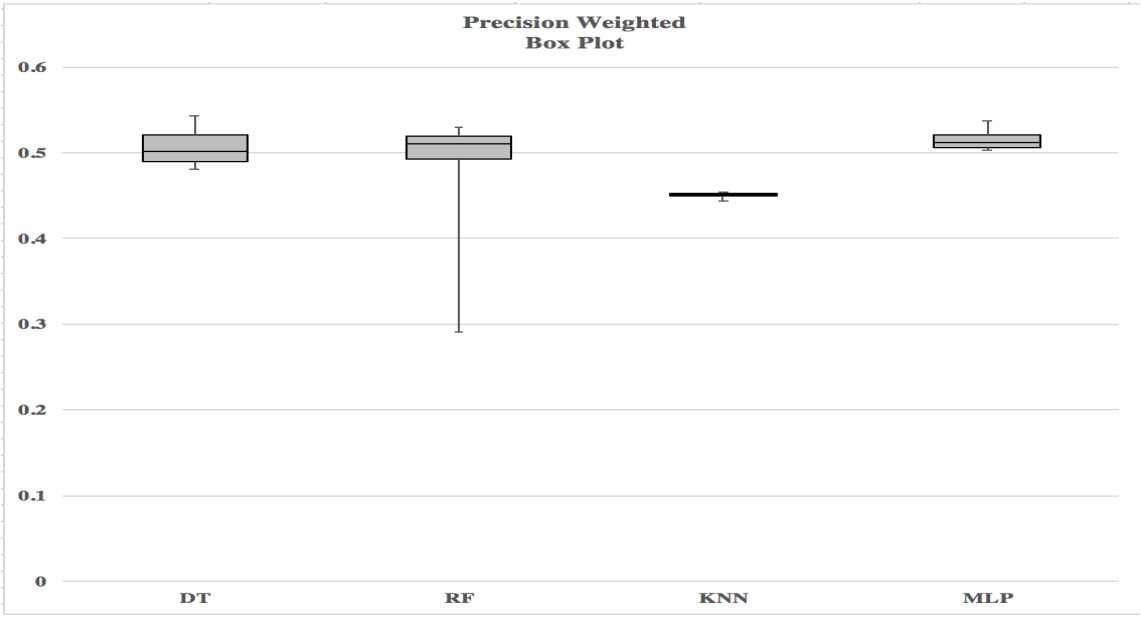


Figure 4.70. Precision Weighted Box Plot for Readmission Dataset Using All Features

Table 4.34. Recall Macro Value for Readmission Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.374854727 | 0.333333333 | 0.34826284 | 0.365324186 |
| First Quartile (Q1) | 0.390672093 | 0.371592365 | 0.351956388 | 0.372574578 |
| Median Value | 0.395362031 | 0.386150699 | 0.35255009 | 0.374607717 |
| Third Quartile(Q3) | 0.397613338 | 0.38867839 | 0.353238117 | 0.378940102 |
| Max Value | 0.399184115 | 0.390136123 | 0.354926916 | 0.389575121 |
| Box 1-hidden (Q1) | 0.390672093 | 0.371592365 | 0.351956388 | 0.372574578 |
| Box 2 (Median - Q1) | 0.004689938 | 0.014558334 | 0.000593702 | 0.002033139 |
| Box 3 (Q3-Median) | 0.002251307 | 0.002527692 | 0.000688026 | 0.004332386 |
| Whisker Top (Max- Q3) | 0.001570778 | 0.001457733 | 0.001688799 | 0.010635019 |
| Whisker Bottom (Q1- Min) | 0.015817366 | 0.038259031 | 0.003693548 | 0.007250392 |

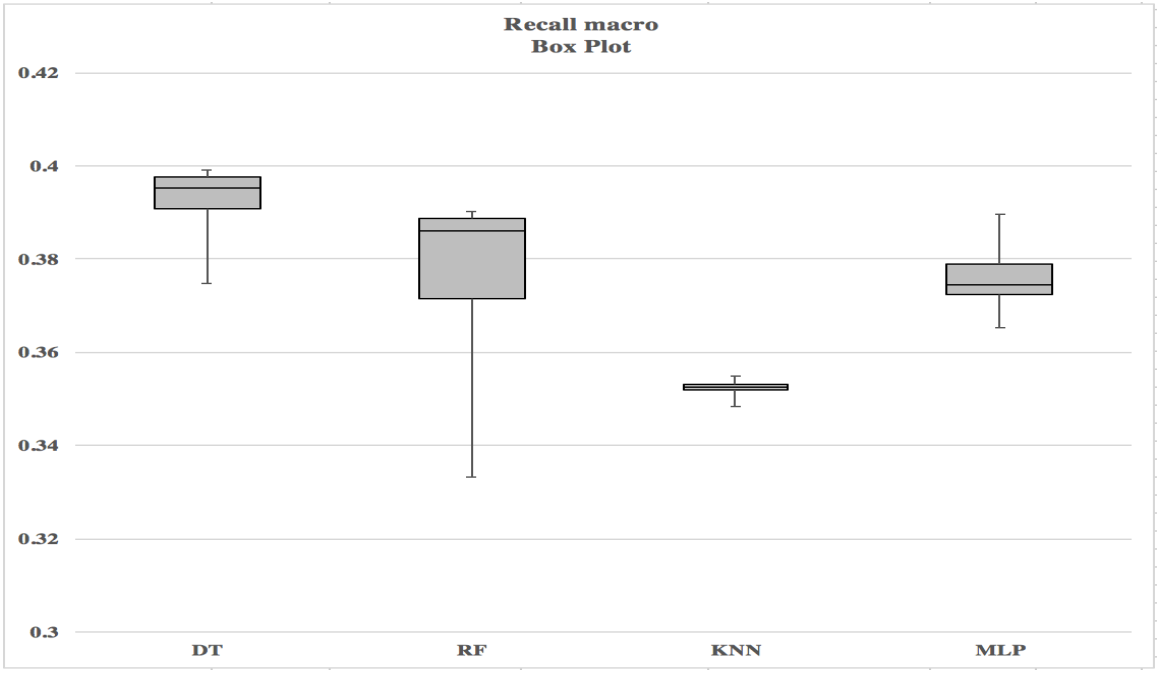


Figure 4.71. Recall Macro Box Plot for Readmission Dataset Using All Features

Table 4.35. Recall Micro Value for Readmission Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.500514501 | 0.539786407 | 0.378462891 | 0.48790673 |
| First Quartile (Q1) | 0.533280218 | 0.552688892 | 0.464574363 | 0.513274624 |
| Median Value | 0.558872893 | 0.560791033 | 0.492831953 | 0.53229867 |
| Third Quartile(Q3) | 0.565306653 | 0.564397534 | 0.502542534 | 0.538530126 |
| Max Value | 0.573843372 | 0.56797906 | 0.508436816 | 0.557908828 |
| Box 1-hidden (Q1) | 0.533280218 | 0.552688892 | 0.464574363 | 0.513274624 |
| Box 2 (Median - Q1) | 0.025592675 | 0.008102141 | 0.02825759 | 0.019024047 |
| Box 3 (Q3- Median) | 0.006433759 | 0.003606502 | 0.009710581 | 0.006231455 |
| Whisker Top (Max- Q3) | 0.008536719 | 0.003581526 | 0.005894283 | 0.019378703 |
| Whisker Bottom (Q1- Min) | 0.032765717 | 0.012902485 | 0.086111472 | 0.025367893 |

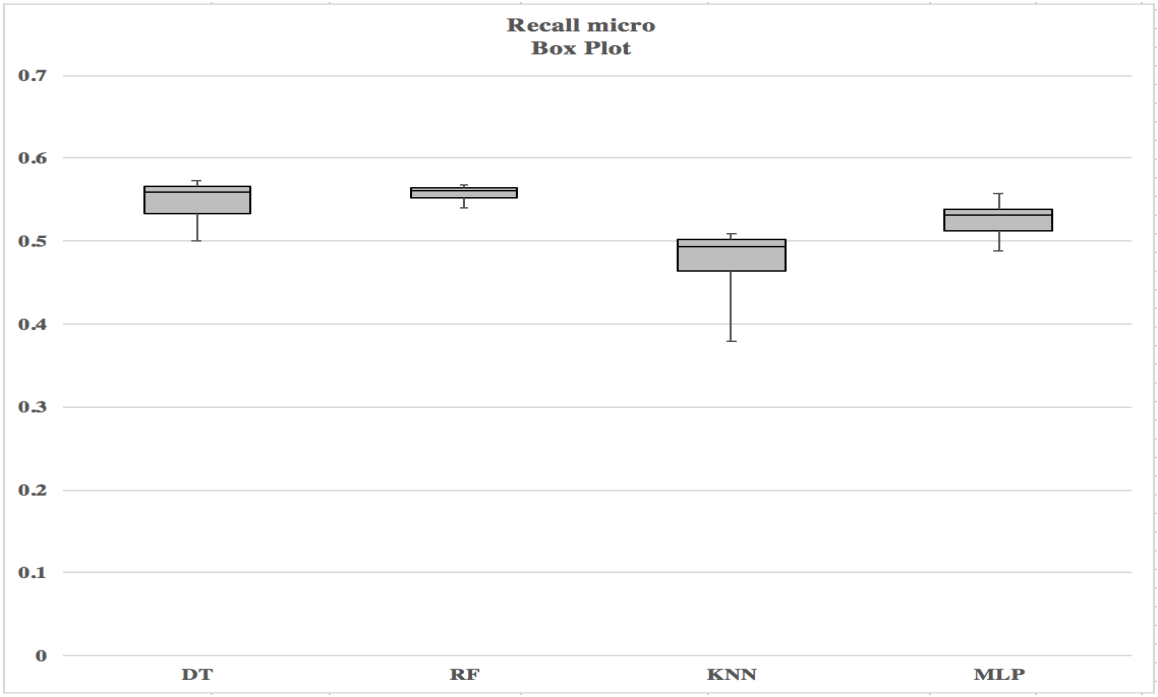


Figure 4.72. Recall Micro Box Plot for Readmission Dataset Using All Features

Table 4.36. Recall Weighted Value for Readmission Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.498936032 | 0.539786407 | 0.378462891 | 0.51274264 |
| First Quartile (Q1) | 0.533217779 | 0.552688892 | 0.464574363 | 0.522038623 |
| Median Value | 0.558817947 | 0.560791033 | 0.492831953 | 0.526938869 |
| Third Quartile(Q3) | 0.565344116 | 0.564397534 | 0.502542534 | 0.536584513 |
| Max Value | 0.573853362 | 0.56797906 | 0.508436816 | 0.551475069 |
| Box 1-hidden (Q1) | 0.533217779 | 0.552688892 | 0.464574363 | 0.522038623 |
| Box 2 (Median - Q1) | 0.025600168 | 0.008102141 | 0.02825759 | 0.004900247 |
| Box 3 (Q3- Median) | 0.00652617 | 0.003606502 | 0.009710581 | 0.009645644 |
| Whisker Top (Max- Q3) | 0.008509246 | 0.003581526 | 0.005894283 | 0.014890556 |
| Whisker Bottom (Q1- Min) | 0.034281747 | 0.012902485 | 0.086111472 | 0.009295983 |

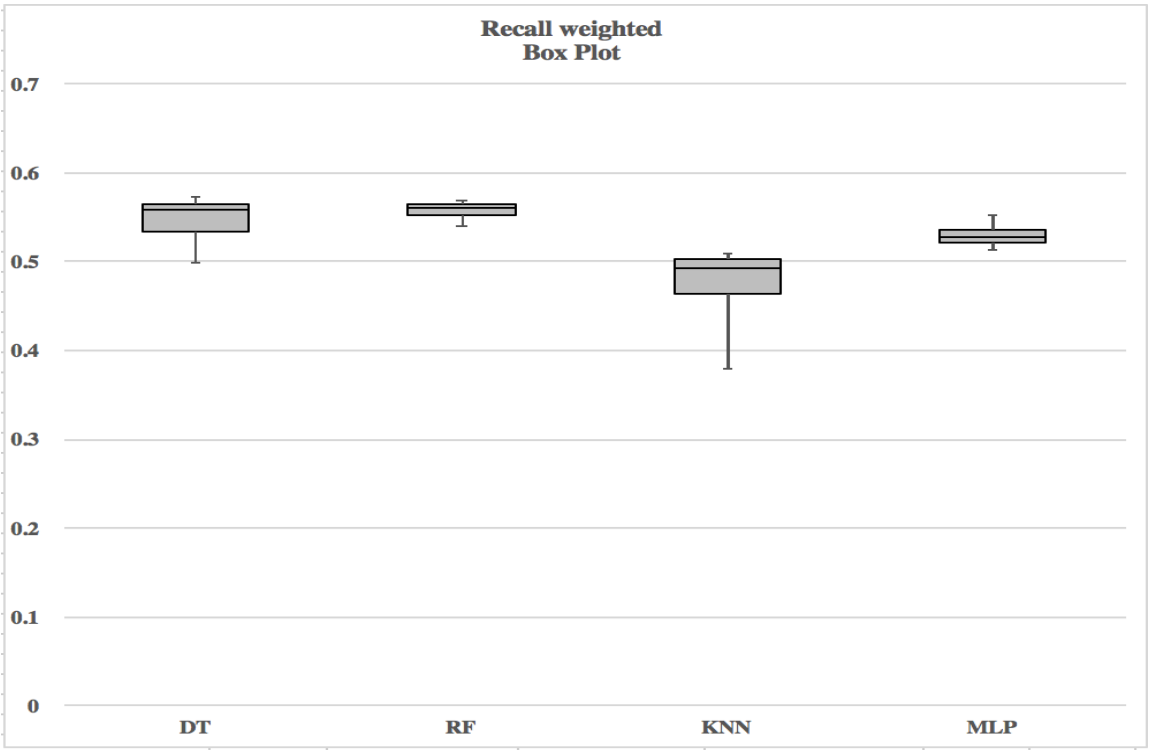


Figure 4.73. Recall Weighted Box Plot for Readmission Dataset Using All Features

#### 4.5.1.3 Best Model

The following diagram shows the best model of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithms using all the features of dataset.

| Algorithm | Accuracy | | Precision Micro | | Precision Macro | | Precision Weighted | | Recall Micro | | Recall Macro | | Recall Weighted | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value |
| Decission Tree | max_depth: 6 | 0.573843372 | max_depth: 6 | 0.573853362 | max_depth: 6 | 0.501806474 | max_depth: 6 | 0.542467919 | max_depth: 6 | 0.573843372 | max_depth: 5 | 0.399184115 | max_depth: 6 | 0.573853362 |
| Random Forest | max_depth: 9 | 0.56797906 | max_depth: 9 | 0.56797906 | max_depth: 11 | 0.475013144 | max_depth: 11 | 0.529759957 | max_depth: 9 | 0.56797906 | max_depth: 17 | 0.390136123 | max_depth: 9 | 0.56797906 |
| KNN | n_neighbors: 20 | 0.508436816 | n_neighbors: 20 | 0.508436816 | n_neighbors: 16 | 0.369646216 | n_neighbors: 13 | 0.453840202 | n_neighbors: 20 | 0.508436816 | n_neighbors: 5 | 0.354926916 | n_neighbors: 20 | 0.508436816 |
| MLP | max_iter: 20000 | 0.553912705 | max_iter: 20000 | 0.550915612 | max_iter: 130000 | 0.474367028 | max_iter: 190000 | 0.536597265 | max_iter: 80000 | 0.557908828 | max_iter: 160000 | 0.389575121 | max_iter: 50000 | 0.551475069 |

Figure 4.74. Best Models for Readmission Dataset Using All Features

### 4.5.2 Using Transfer Learning

In transfer learning technique, top 10 important features were identified during transfer learning using decision tree. And these top 10 features were only used for training all the models

of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithm to demonstrate the transfer learning in this experiment. The following sections shows the comparison between the models.

#### 4.5.2.1 Line Graph

Following line diagram shows the comparison of different models of each algorithm based on evaluation metrics.

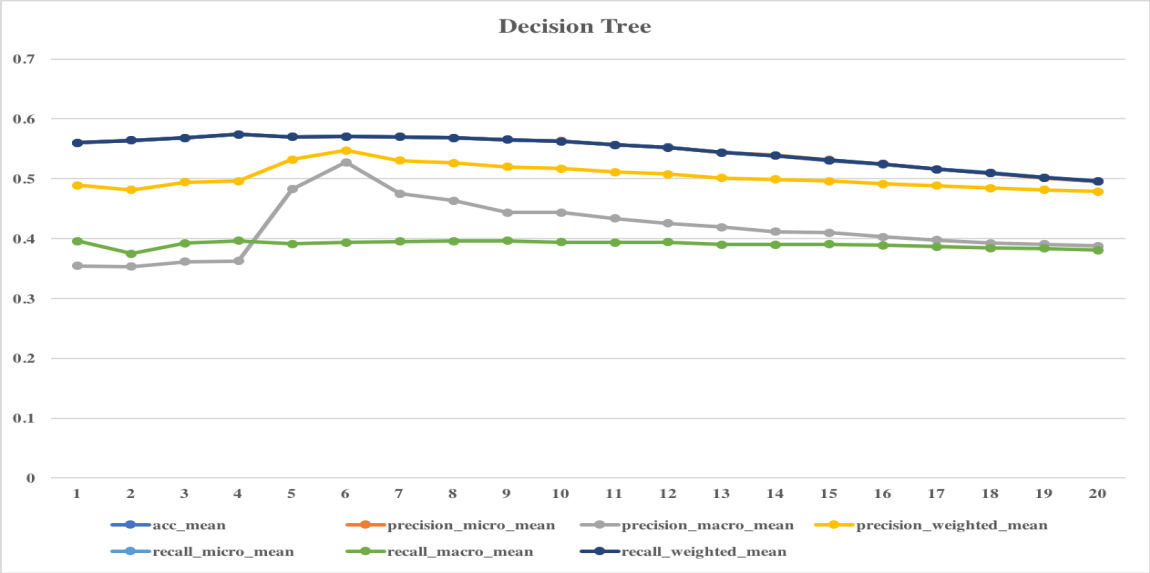


Figure 4.75. Line Graph of Decision Tree with Varying Max_Depth for Readmission Dataset Using Transfer Learning

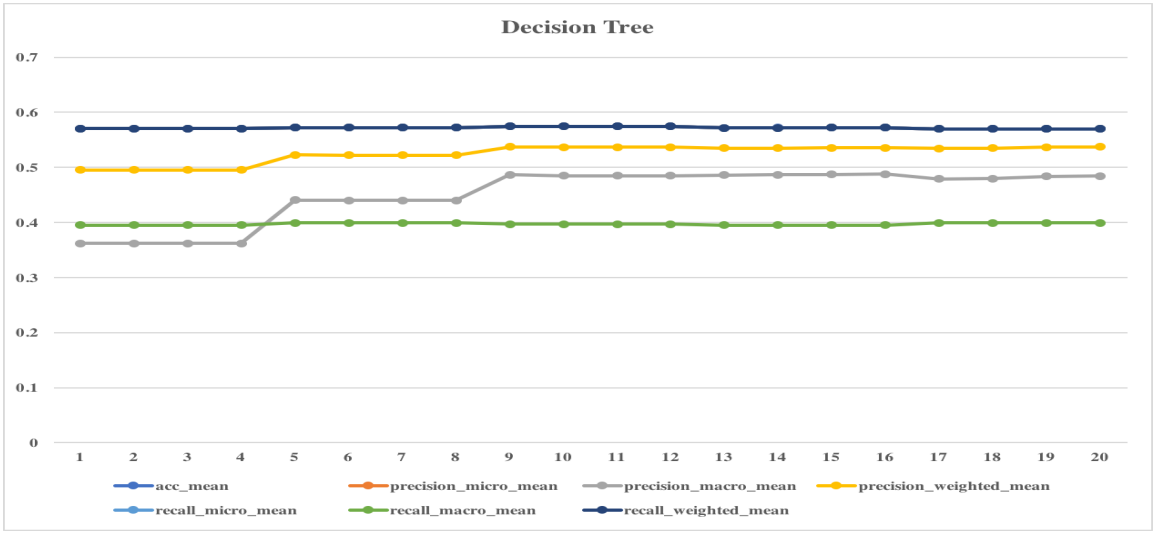


Figure 4.76. Line Graph of Decision Tree with Varying Max_Depth and Min_Sample_Split for Readmission Dataset Using Transfer Learning

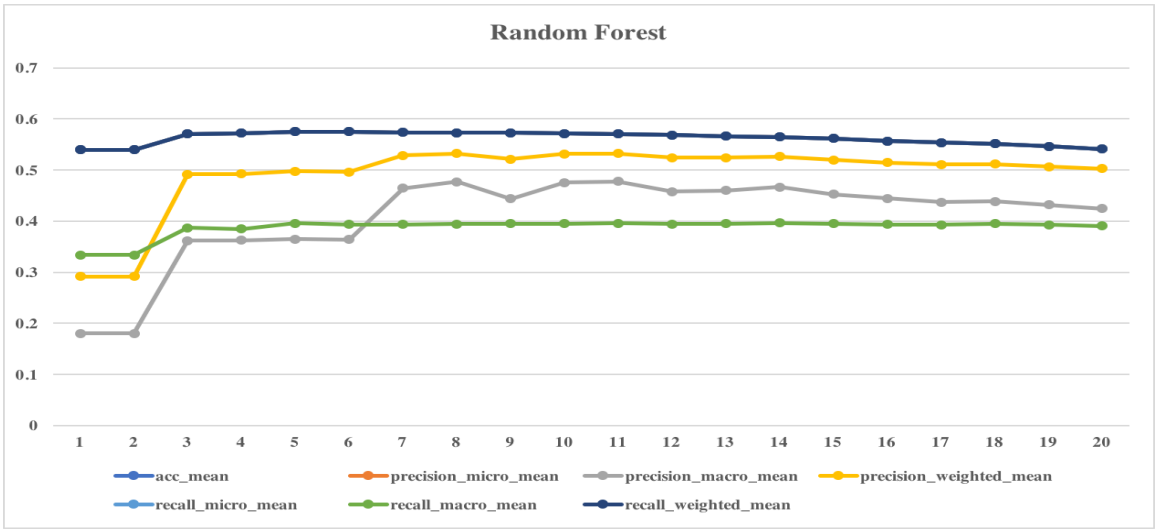


Figure 4.77. Line Graph of Random Forest with Varying Max_Depth for Readmission Dataset Using Transfer Learning

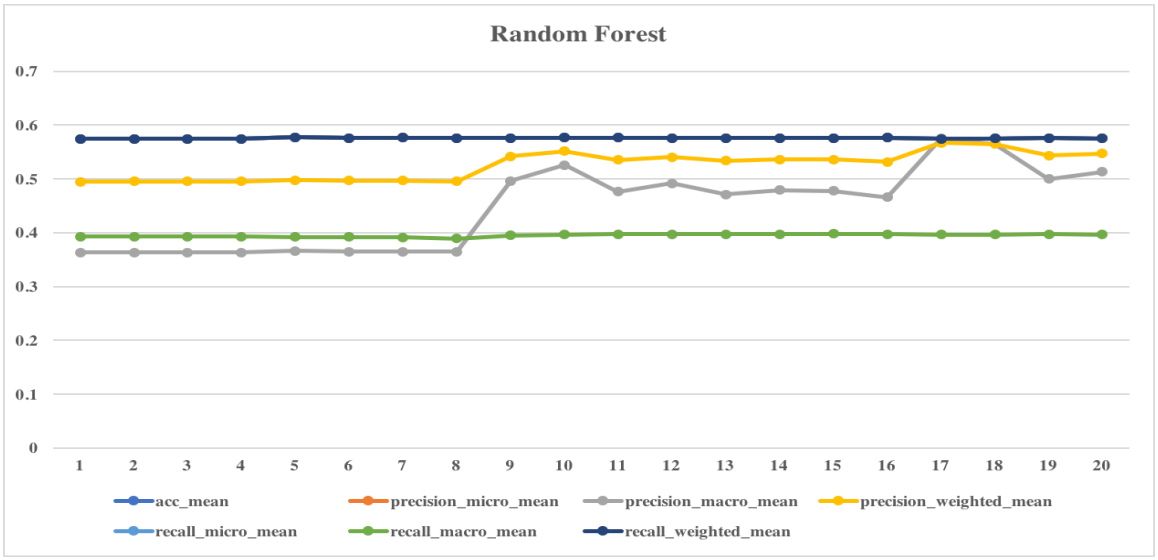


Figure 4.78. Line Graph of Random Forest with Varying Max_Depth and Min_Sample_Split for Readmission Dataset Using Transfer Learning

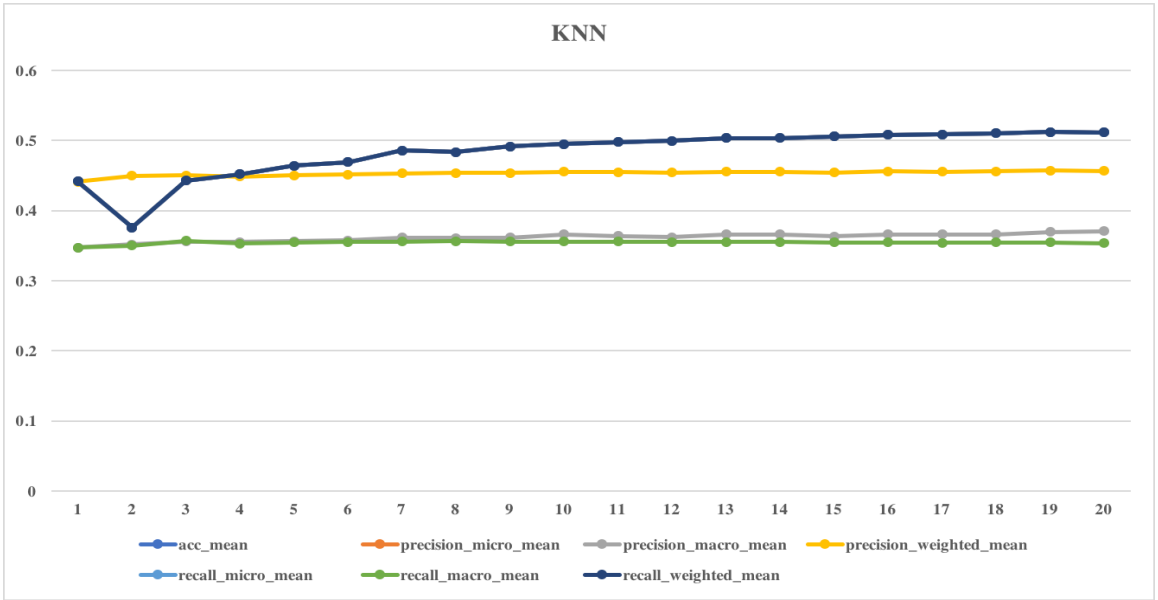


Figure 4.79. Line Graph of KNN with Varying N_Neighbor for Readmission Dataset Using Transfer Learning

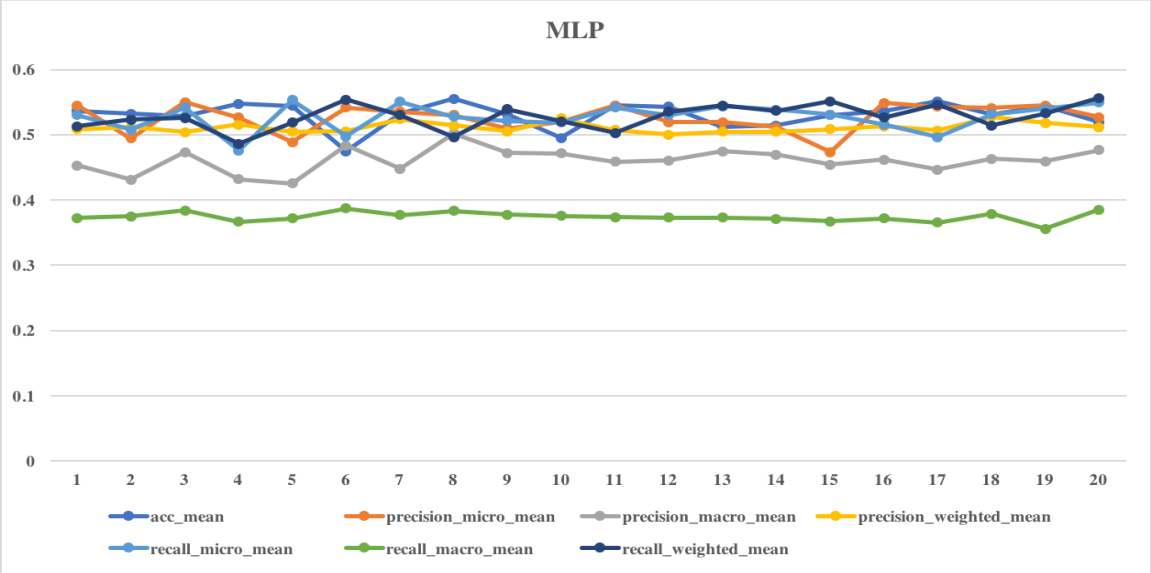


Figure 4.80. Line Graph of MLP with Varying Max_Iteration for Readmission Dataset Using Transfer Learning.

#### 4.5.2.2 Box Plot

Following box diagram shows the comparison of each algorithm based on grid search with 5-fold cross validation.

Table 4.37. Accuracy Value for Readmission Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.495069782 | 0.539786407 | 0.375875401 | 0.47433989 |
| First Quartile (Q1) | 0.528941926 | 0.553048543 | 0.467881155 | 0.525877399 |
| Median Value | 0.558313436 | 0.567574453 | 0.496403489 | 0.532378593 |
| Third Quartile(Q3) | 0.567874162 | 0.572374796 | 0.506543653 | 0.543720092 |
| Max Value | 0.573733479 | 0.575461802 | 0.512183182 | 0.554671968 |
| Box 1-hidden (Q1) | 0.528941926 | 0.553048543 | 0.467881155 | 0.525877399 |
| Box 2 (Median - Q1) | 0.02937151 | 0.01452591 | 0.028522333 | 0.006501194 |
| Box 3 (Q3-Median) | 0.009560726 | 0.004800344 | 0.010140164 | 0.011341499 |
| Whisker Top (Max- Q3) | 0.005859316 | 0.003087006 | 0.00563953 | 0.010951877 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Whisker Bottom (Q1- Min) | 0.033872144 | 0.013262136 | 0.092005754 | 0.051537509 |

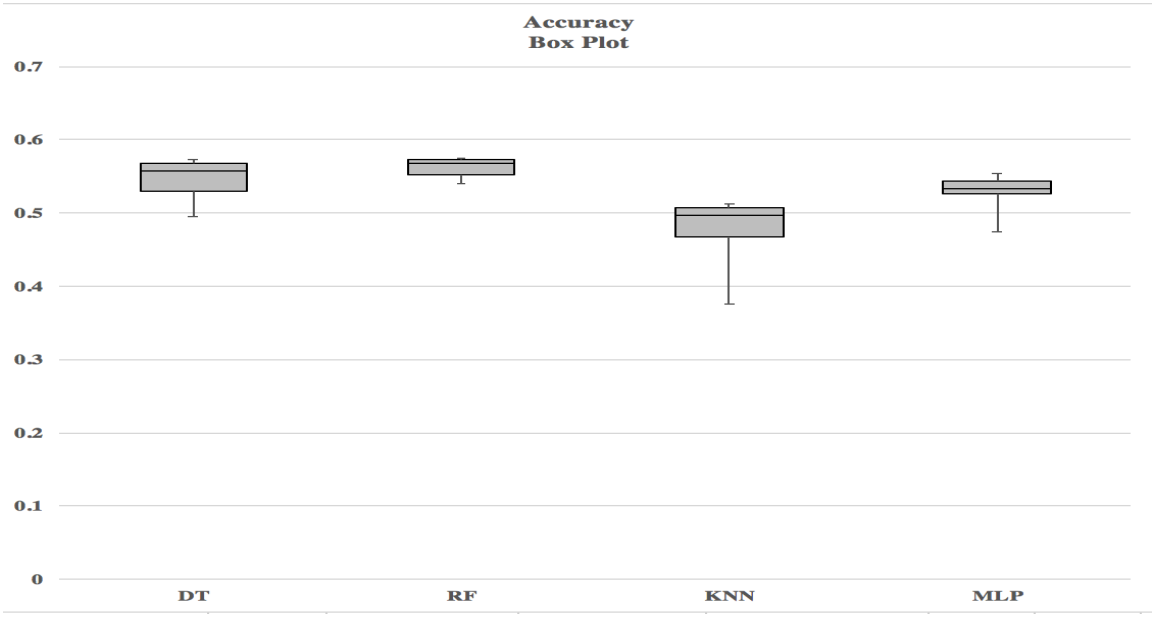


Figure 4.81. Accuracy Box Plot for Readmission Dataset Using Transfer Learning

Table 4.38. Precision Macro Value for Readmission Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.353411859 | 0.179928802 | 0.347604206 | 0.425288725 |
| First Quartile (Q1) | 0.389470753 | 0.364414957 | 0.357282973 | 0.451741073 |
| Median Value | 0.410736496 | 0.441054535 | 0.362951623 | 0.461277677 |
| Third Quartile(Q3) | 0.443289847 | 0.461292197 | 0.366261686 | 0.472450464 |
| Max Value | 0.527399005 | 0.477369725 | 0.370851728 | 0.501755071 |
| Box 1-hidden (Q1) | 0.389470753 | 0.364414957 | 0.357282973 | 0.451741073 |
| Box 2 (Median - Q1) | 0.021265743 | 0.076639578 | 0.00566865 | 0.009536604 |
| Box 3 (Q3- Median) | 0.032553351 | 0.020237662 | 0.003310063 | 0.011172787 |
| Whisker Top (Max- Q3) | 0.084109159 | 0.016077528 | 0.004590042 | 0.029304607 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Whisker Bottom (Q1- Min) | 0.036058893 | 0.184486155 | 0.009678767 | 0.026452348 |

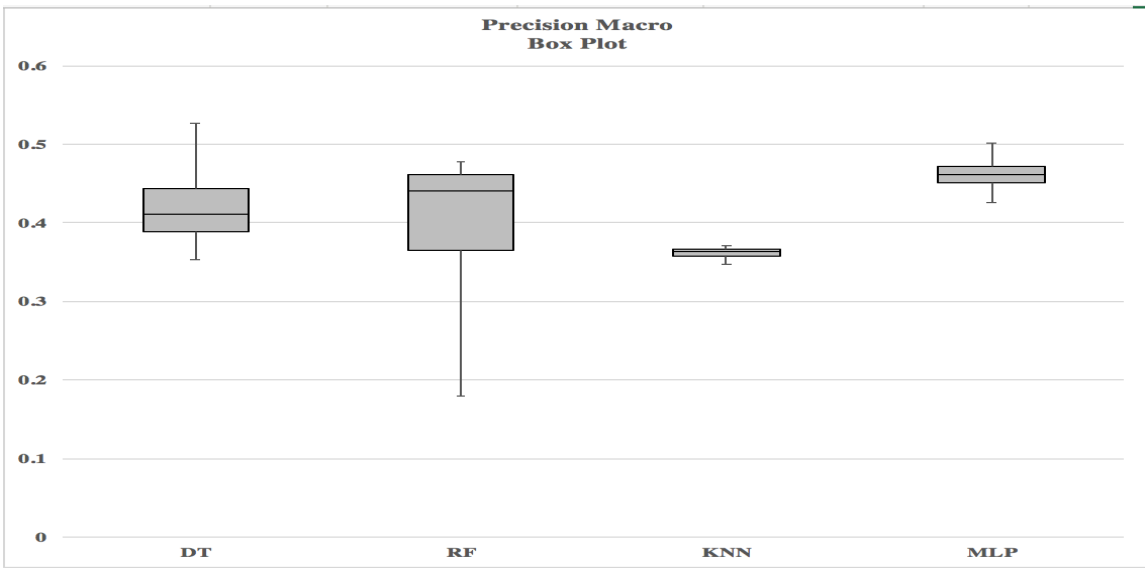


Figure 4.82. Precision Macro Box Plot for Readmission Dataset Using Transfer Learning

Table 4.39. Precision Micro Value for Readmission Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.495389472 | 0.539786407 | 0.375875401 | 0.473490714 |
| First Quartile (Q1) | 0.529608779 | 0.553048543 | 0.467881155 | 0.517100912 |
| Median Value | 0.558263484 | 0.567574453 | 0.496403489 | 0.52840245 |
| Third Quartile(Q3) | 0.567966572 | 0.572374796 | 0.506543653 | 0.543415387 |
| Max Value | 0.573733479 | 0.575461802 | 0.512183182 | 0.550166339 |
| Box 1-hidden (Q1) | 0.529608779 | 0.553048543 | 0.467881155 | 0.517100912 |
| Box 2 (Median - Q1) | 0.028654705 | 0.01452591 | 0.028522333 | 0.011301538 |
| Box 3 (Q3- Median) | 0.009703088 | 0.004800344 | 0.010140164 | 0.015012937 |
| Whisker Top (Max- Q3) | 0.005766906 | 0.003087006 | 0.00563953 | 0.006750952 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Whisker Bottom (Q1- Min) | 0.034219307 | 0.013262136 | 0.092005754 | 0.043610198 |

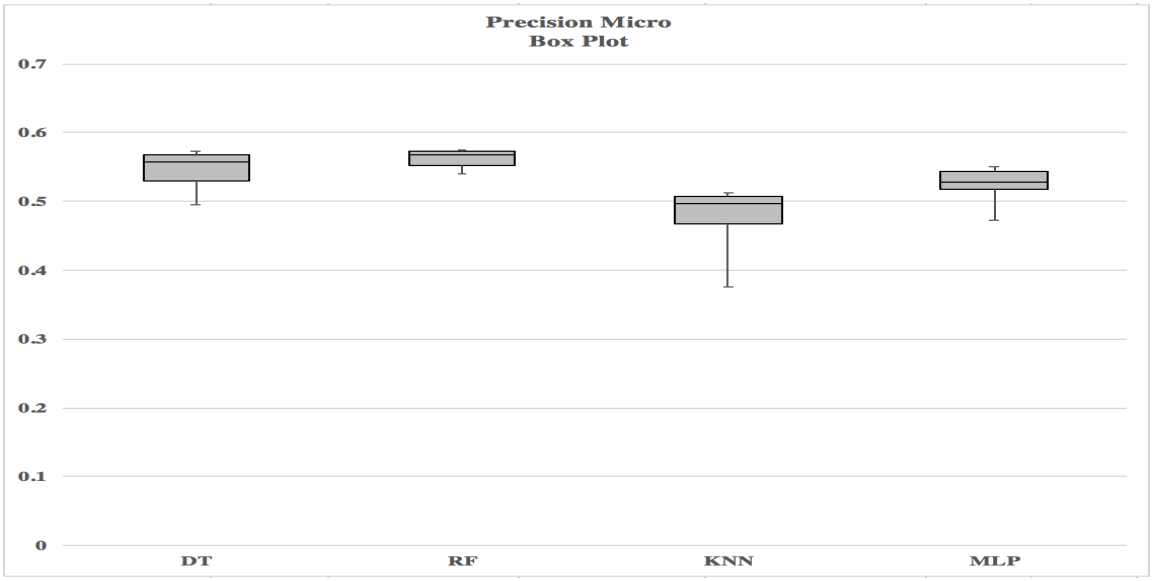


Figure 4.83. Precision Micro Box Plot for Readmission Dataset Using Transfer Learning

Table 4.40. Precision Weighted Value for Readmission Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.478093696 | 0.291369365 | 0.441252654 | 0.500029561 |
| First Quartile (Q1) | 0.488720513 | 0.497107327 | 0.451355489 | 0.504884877 |
| Median Value | 0.497296359 | 0.513304049 | 0.454360988 | 0.508306486 |
| Third Quartile(Q3) | 0.517572402 | 0.524913013 | 0.455516826 | 0.514515907 |
| Max Value | 0.54720419 | 0.532668117 | 0.457151465 | 0.527490462 |
| Box 1-hidden (Q1) | 0.488720513 | 0.497107327 | 0.451355489 | 0.504884877 |
| Box 2 (Median - Q1) | 0.008575846 | 0.016196722 | 0.003005499 | 0.003421609 |
| Box 3 (Q3- Median) | 0.020276043 | 0.011608963 | 0.001155838 | 0.006209421 |
| Whisker Top (Max- Q3) | 0.029631788 | 0.007755104 | 0.001634639 | 0.012974555 |
| Whisker Bottom (Q1- Min) | 0.010626817 | 0.205737962 | 0.010102835 | 0.004855315 |

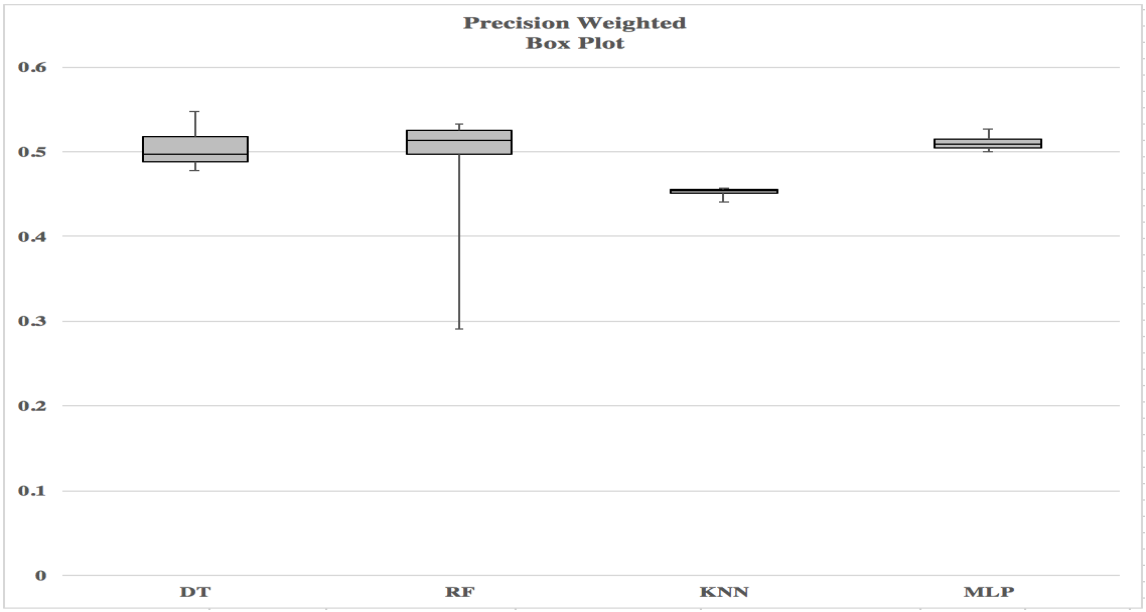


Figure 4.84. Precision Weighted Box Plot for Readmission Dataset Using Transfer Learning

Table 4.41. Recall Macro Value for Readmission Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.374854727 | 0.333333333 | 0.347455826 | 0.355774615 |
| First Quartile (Q1) | 0.388194396 | 0.392109373 | 0.354527354 | 0.371333938 |
| Median Value | 0.391734105 | 0.393829159 | 0.355063449 | 0.37322024 |
| Third Quartile(Q3) | 0.394480788 | 0.394951492 | 0.356015554 | 0.377626216 |
| Max Value | 0.39621497 | 0.396523041 | 0.356949399 | 0.386663989 |
| Box 1-hidden (Q1) | 0.388194396 | 0.392109373 | 0.354527354 | 0.371333938 |
| Box 2 (Median - Q1) | 0.00353971 | 0.001719786 | 0.000536095 | 0.001886302 |
| Box 3 (Q3- Median) | 0.002746682 | 0.001122333 | 0.000952105 | 0.004405976 |
| Whisker Top (Max- Q3) | 0.001734182 | 0.001571549 | 0.000933845 | 0.009037773 |
| Whisker Bottom (Q1- Min) | 0.013339669 | 0.058776039 | 0.007071528 | 0.015559324 |

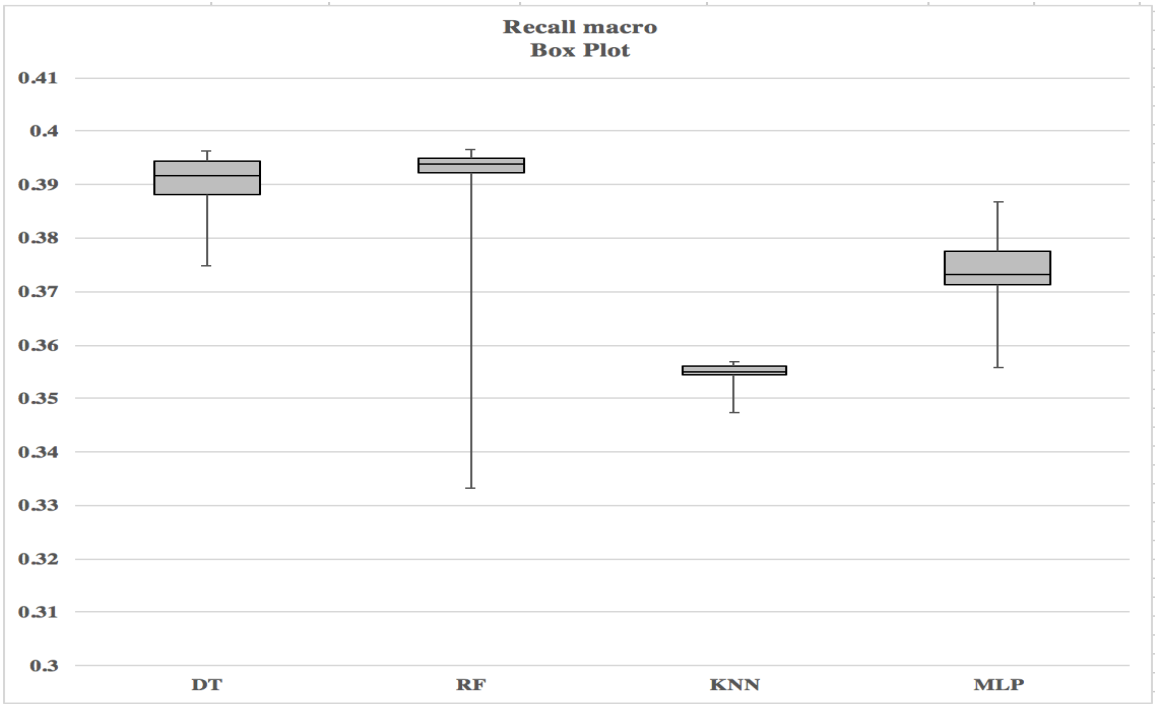


Figure 4.85. Recall Macro Box Plot for Readmission Dataset Using Transfer Learning

Table 4.42. Recall Micro Value for Readmission Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.495109744 | 0.539786407 | 0.375875401 | 0.475558708 |
| First Quartile (Q1) | 0.529029342 | 0.553048543 | 0.467881155 | 0.517725306 |
| Median Value | 0.558378373 | 0.567574453 | 0.496403489 | 0.530565352 |
| Third Quartile(Q3) | 0.567919118 | 0.572374796 | 0.506543653 | 0.541514731 |
| Max Value | 0.573733479 | 0.575461802 | 0.512183182 | 0.552853732 |
| Box 1-hidden (Q1) | 0.529029342 | 0.553048543 | 0.467881155 | 0.517725306 |
| Box 2 (Median - Q1) | 0.029349031 | 0.01452591 | 0.028522333 | 0.012840045 |
| Box 3 (Q3- Median) | 0.009540745 | 0.004800344 | 0.010140164 | 0.010949379 |
| Whisker Top (Max- Q3) | 0.00581436 | 0.003087006 | 0.00563953 | 0.011339001 |
| Whisker Bottom (Q1- Min) | 0.033919598 | 0.013262136 | 0.092005754 | 0.042166598 |

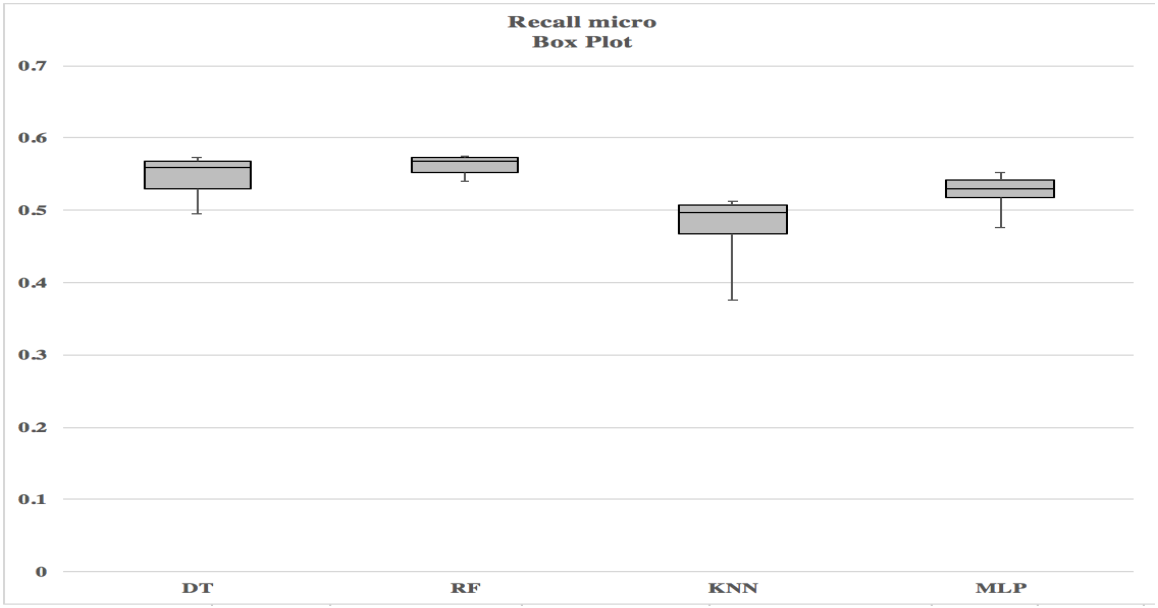


Figure 4.86. Recall Micro Box Plot for Readmission Dataset Using Transfer Learning

Table 4.43. Recall Weighted Value for Readmission Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.495689181 | 0.539786407 | 0.375875401 | 0.486248339 |
| First Quartile (Q1) | 0.529306573 | 0.553048543 | 0.467881155 | 0.517375646 |
| Median Value | 0.558333417 | 0.567574453 | 0.496403489 | 0.527922915 |
| Third Quartile(Q3) | 0.567919118 | 0.572374796 | 0.506543653 | 0.540490724 |
| Max Value | 0.573733479 | 0.575461802 | 0.512183182 | 0.55603065 |
| Box 1-hidden (Q1) | 0.529306573 | 0.553048543 | 0.467881155 | 0.517375646 |
| Box 2 (Median - Q1) | 0.029026844 | 0.01452591 | 0.028522333 | 0.010547269 |
| Box 3 (Q3- Median) | 0.009585702 | 0.004800344 | 0.010140164 | 0.012567809 |
| Whisker Top (Max- Q3) | 0.00581436 | 0.003087006 | 0.00563953 | 0.015539926 |
| Whisker Bottom (Q1- Min) | 0.033617391 | 0.013262136 | 0.092005754 | 0.031127307 |

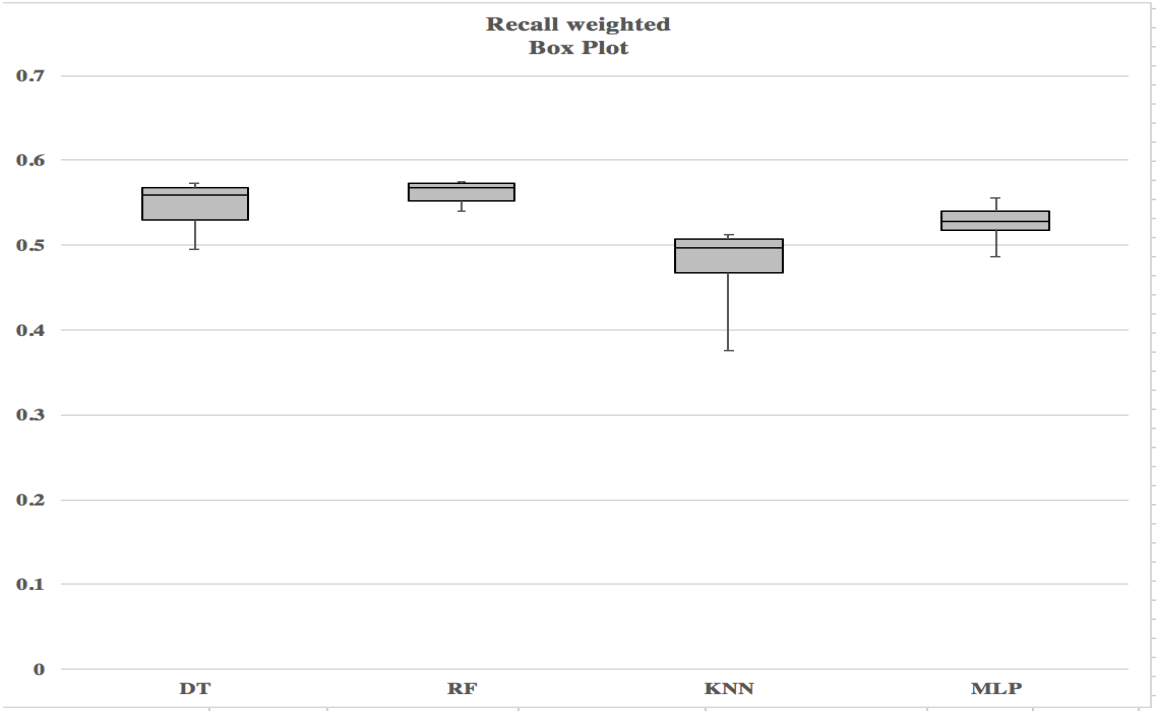


Figure 4.87. Recall Weighted Box Plot for Readmission Dataset Using Transfer Learning

#### 4.5.2.3 Best Model

The following diagram shows the best model of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithms using transfer learning.

| Algorithm | Accuracy | | Precision Micro | | Precision Macro | | Precision Weighted | | Recall Micro | | Recall Macro | | Recall Weighted | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value |
| Decission Tree | max_depth: 4 | 0.573733479 | max_depth: 4 | 0.573733479 | max_depth: 6 | 0.527399005 | max_depth: 6 | 0.54720419 | max_depth: 4 | 0.573733479 | max_depth: 9 | 0.39621497 | max_depth: 4 | 0.573733479 |
| Random Forest | max_depth: 6 | 0.575461802 | max_depth: 6 | 0.575461802 | max_depth: 11 | 0.477369725 | max_depth: 8 | 0.532668117 | max_depth: 6 | 0.575461802 | max_depth: 14 | 0.396523041 | max_depth: 6 | 0.575461802 |
| KNN | n_neighbors: 19 | 0.512183182 | n_neighbors: 19 | 0.512183182 | n_neighbors: 20 | 0.370851728 | n_neighbors: 19 | 0.457151465 | n_neighbors: 19 | 0.512183182 | n_neighbors: 3 | 0.356949399 | n_neighbors: 19 | 0.512183182 |
| MLP | max_iter: 80000 | 0.554671968 | max_iter: 30000 | 0.550166339 | max_iter: 80000 | 0.501755071 | max_iter: 180000 | 0.527490462 | max_iter: 50000 | 0.552853732 | max_iter: 60000 | 0.386663989 | max_iter: 200000 | 0.55603065 |

Figure 4.88. Best Models for Readmission Dataset Using Transfer Learning

### 4.5.3 Using Suggested Feature Technique

In suggested feature technique, all the expert suggested features as specified in section 3.3 were only used for training all the models of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithms. The following sections shows the comparison between the models.

#### 4.5.3.1 Line Graph

Following line diagram shows the comparison of different models of each algorithm based on evaluation metrics.

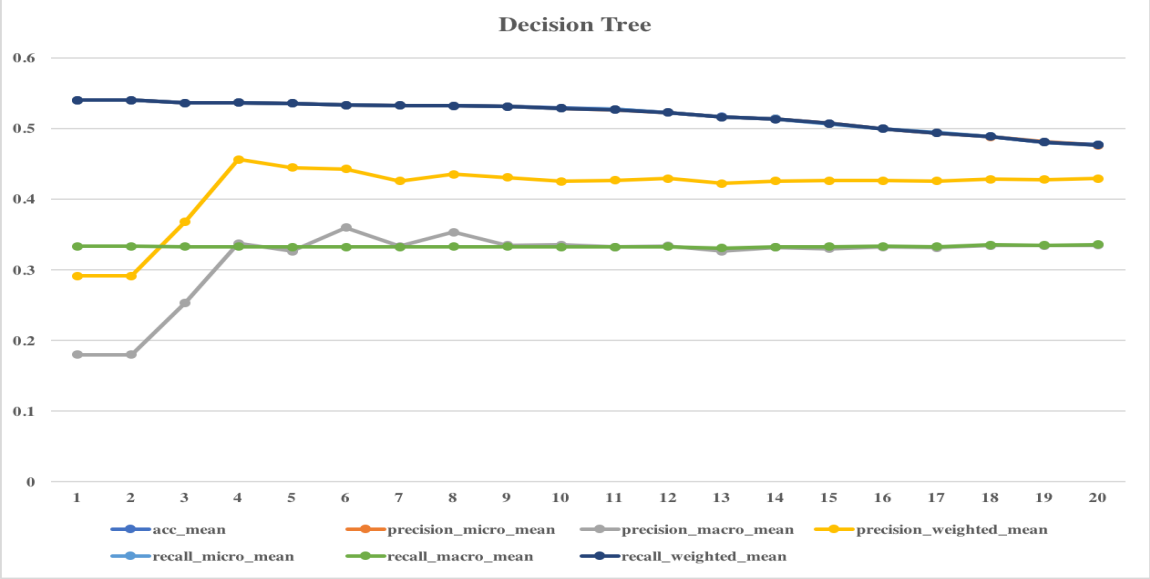


Figure 4.89. Line Graph of Decision Tree with Varying Max_Depth for Readmission Dataset Using Suggested Features

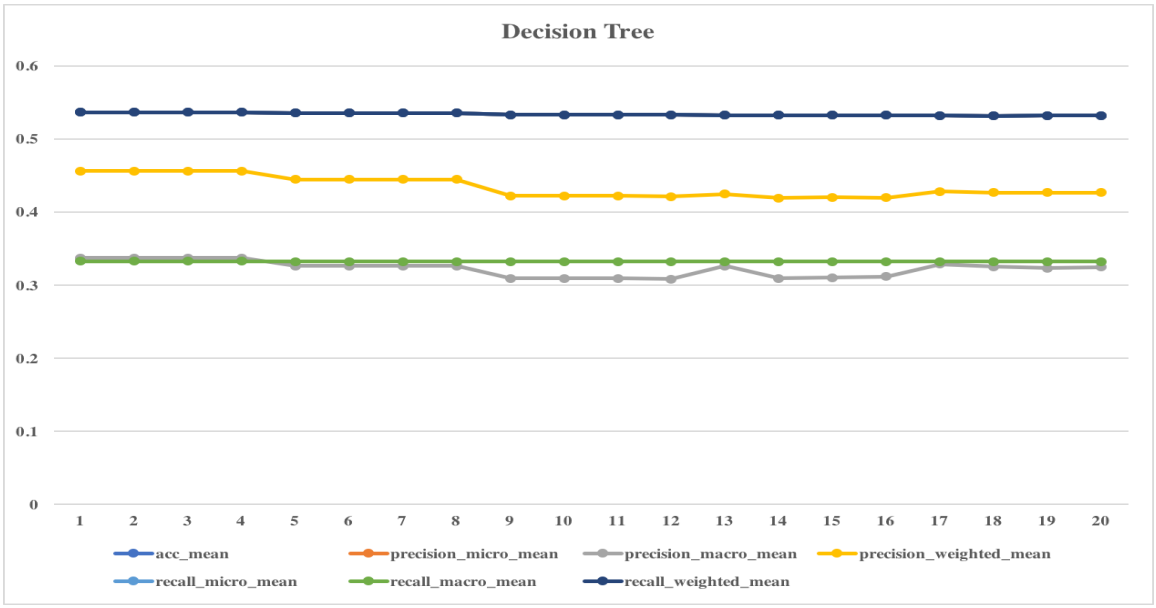


Figure 4.90. Line Graph of Decision Tree with Varying Max_Depth and Min_Sample_Split for Readmission Dataset Using Suggested Features

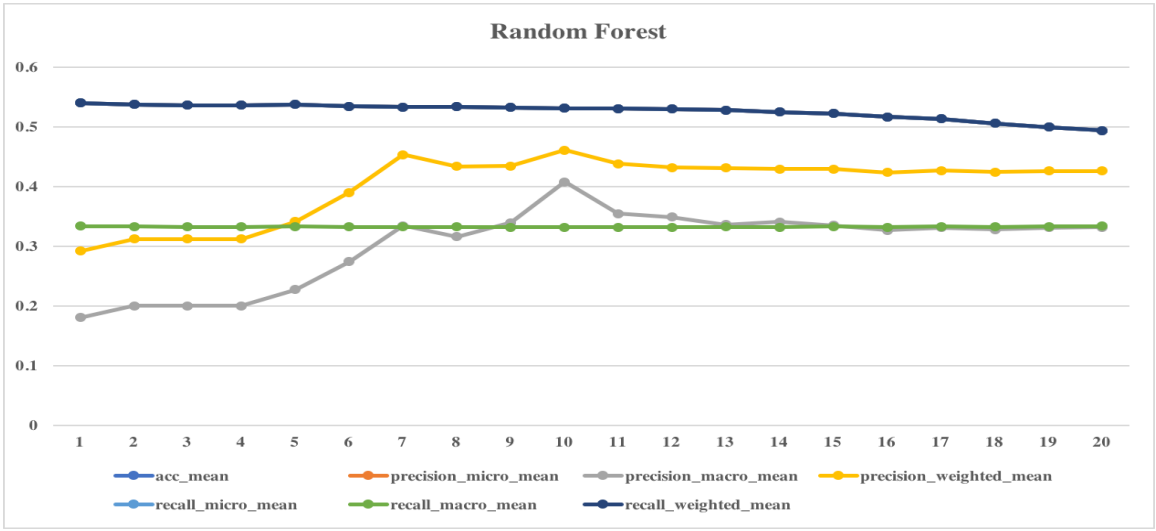


Figure 4.91. Line Graph of Random Forest with Varying Max_Depth for Readmission Dataset Using Suggested Features

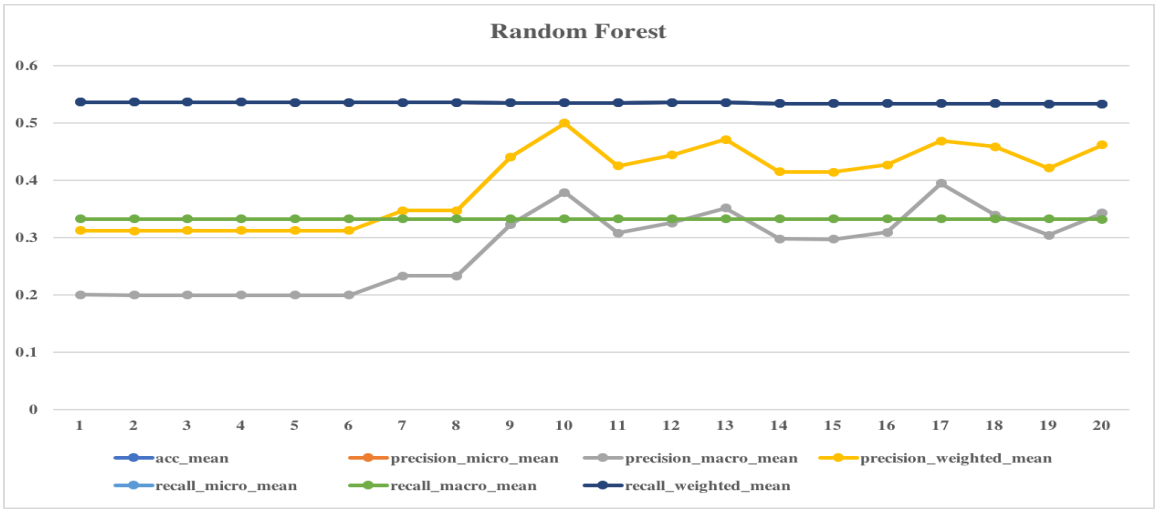


Figure 4.92. Line Graph of Random Forest with Varying Max_Depth and Min_Sample_Split for Readmission Dataset Using Suggested Features.

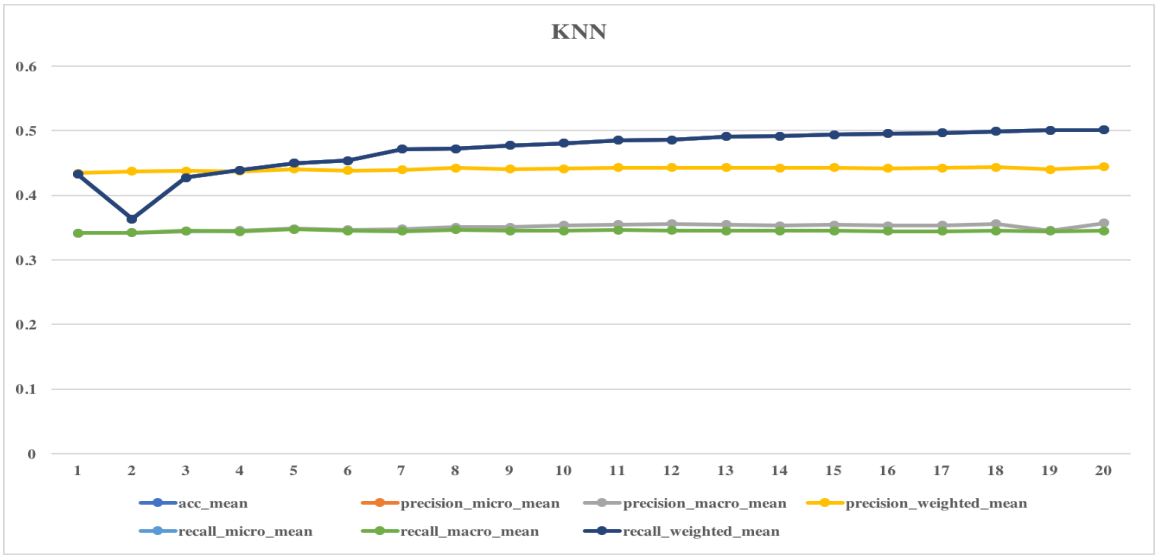


Figure 4.93. Line Graph of KNN with Varying N_Neighbor for Readmission Dataset Using Suggested Features

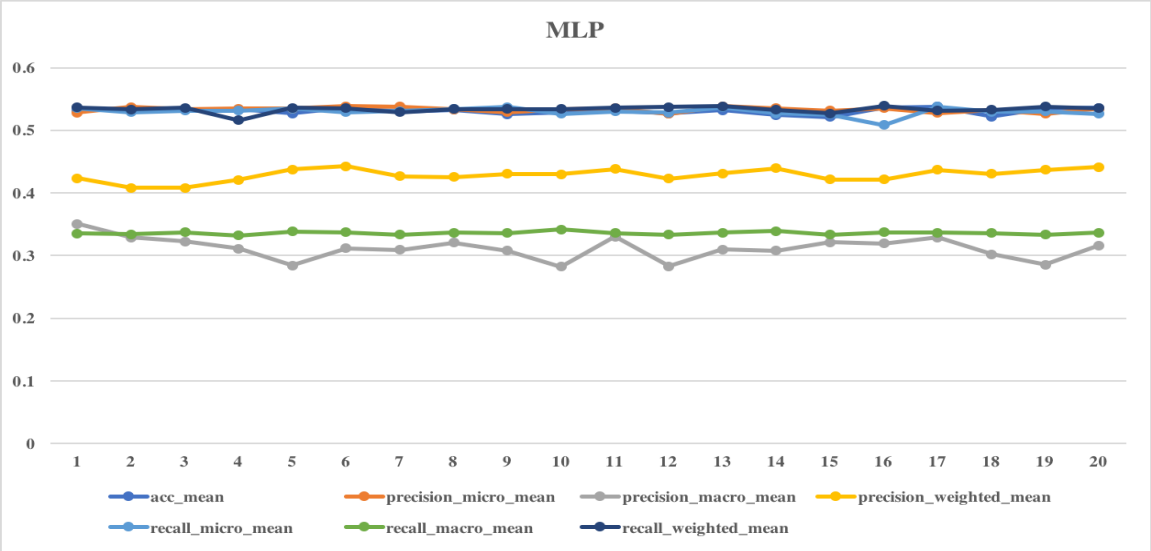


Figure 4.94. Line Graph of MLP with Varying Max_Iteration for Readmission Dataset Using Suggested Features

#### 4.5.3.2 Box Plot

Following box diagram, shows the comparison of each algorithm based on grid search with 5-fold cross validation.

Table 4.44. Accuracy Value for Readmission Dataset Using Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.4769074 | 0.493860955 | 0.363447456 | 0.521334306 |
| First Quartile (Q1) | 0.505050101 | 0.520744877 | 0.452925662 | 0.527043768 |
| Median Value | 0.52776307 | 0.530944983 | 0.482861624 | 0.53197898 |
| Third Quartile(Q3) | 0.533475029 | 0.534376655 | 0.494622716 | 0.533779734 |
| Max Value | 0.539786407 | 0.539786407 | 0.501653396 | 0.537698433 |
| Box 1-hidden (Q1) | 0.505050101 | 0.520744877 | 0.452925662 | 0.527043768 |
| Box 2 (Median - Q1) | 0.022712968 | 0.010200106 | 0.029935962 | 0.004935213 |
| Box 3 (Q3-Median) | 0.005711959 | 0.003431671 | 0.011761092 | 0.001800753 |
| Whisker Top (Max- Q3) | 0.006311378 | 0.005409753 | 0.00703068 | 0.003918699 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Whisker Bottom (Q1- Min) | 0.028142702 | 0.026883923 | 0.089478206 | 0.005709462 |

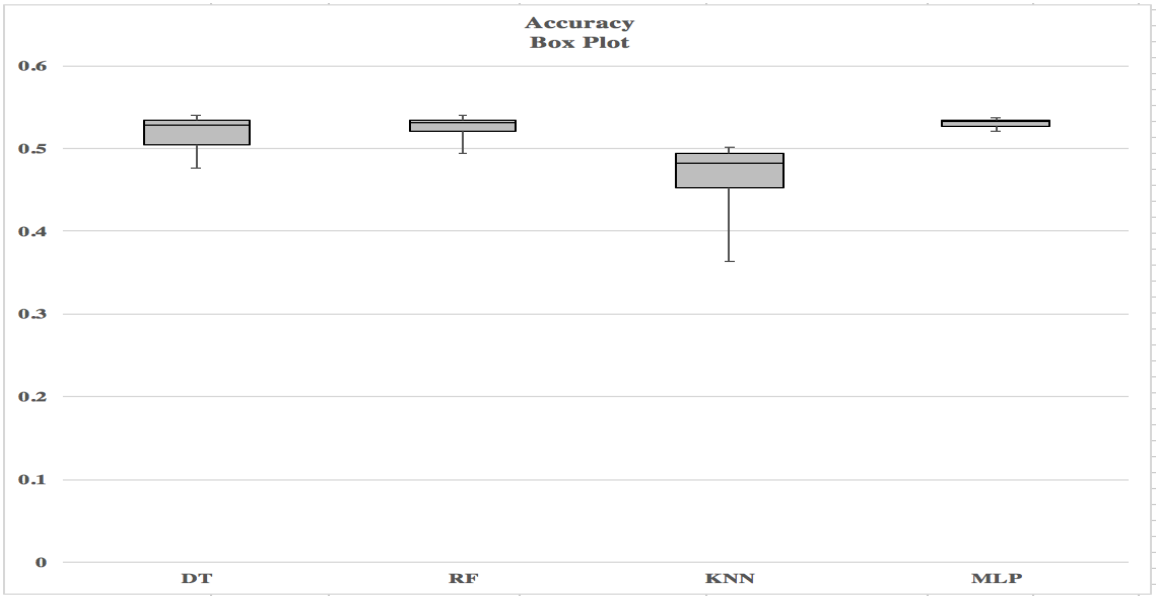


Figure 4.95. Accuracy Box Plot for Readmission Dataset Using Suggested Features

Table 4.45. Precision Macro Value for Readmission Dataset Using Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.179928802 | 0.179928802 | 0.341336542 | 0.282424114 |
| First Quartile (Q1) | 0.328851758 | 0.262241117 | 0.34659749 | 0.306310095 |
| Median Value | 0.332766206 | 0.330828429 | 0.351996591 | 0.311191827 |
| Third Quartile(Q3) | 0.334555954 | 0.33661305 | 0.35439623 | 0.321196042 |
| Max Value | 0.35966497 | 0.407754946 | 0.357126197 | 0.350645358 |
| Box 1-hidden (Q1) | 0.328851758 | 0.262241117 | 0.34659749 | 0.306310095 |
| Box 2 (Median - Q1) | 0.003914449 | 0.068587313 | 0.005399101 | 0.004881732 |
| Box 3 (Q3- Median) | 0.001789747 | 0.005784621 | 0.002399639 | 0.010004214 |
| Whisker Top (Max- Q3) | 0.025109017 | 0.071141896 | 0.002729968 | 0.029449316 |
| Whisker Bottom (Q1- Min) | 0.148922955 | 0.082312314 | 0.005260947 | 0.023885981 |

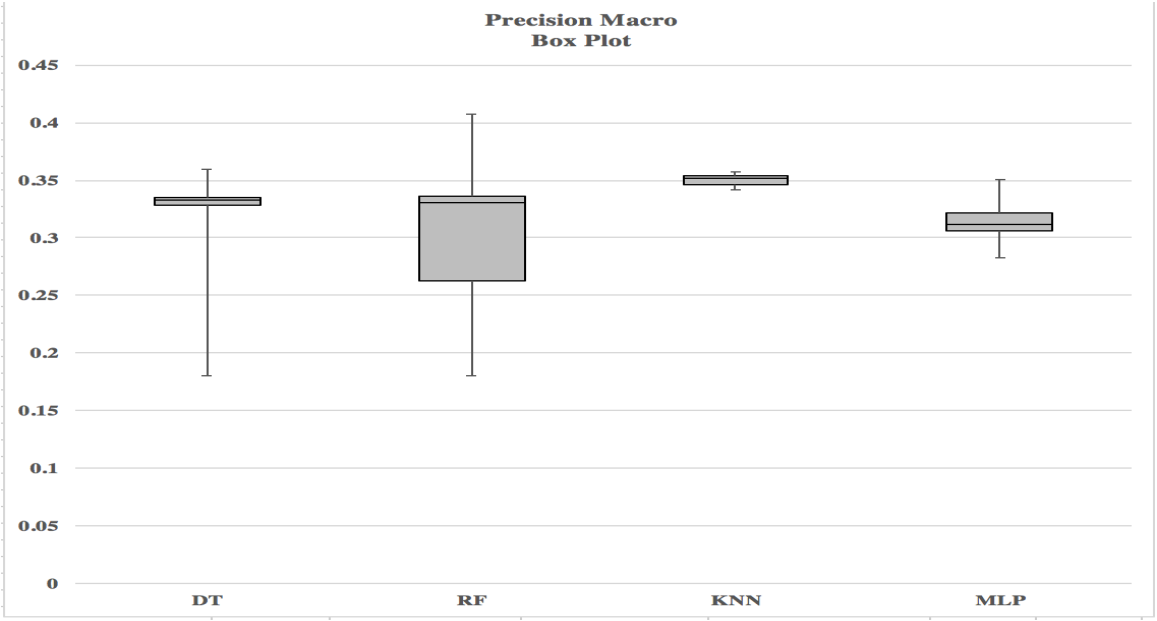


Figure 4.96. Precision Macro Box Plot for Readmission Dataset Using Suggested Features

Table 4.46. Precision Micro Value for Readmission Dataset Using Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.476038243 | 0.493860955 | 0.363447456 | 0.526369422 |
| First Quartile (Q1) | 0.505189966 | 0.520744877 | 0.452925662 | 0.530677743 |
| Median Value | 0.52745337 | 0.530944983 | 0.482861624 | 0.534111911 |
| Third Quartile(Q3) | 0.533475029 | 0.534376655 | 0.494622716 | 0.535130923 |
| Max Value | 0.539786407 | 0.539786407 | 0.501653396 | 0.53859756 |
| Box 1-hidden (Q1) | 0.505189966 | 0.520744877 | 0.452925662 | 0.530677743 |
| Box 2 (Median - Q1) | 0.022263404 | 0.010200106 | 0.029935962 | 0.003434169 |
| Box 3 (Q3- Median) | 0.006021659 | 0.003431671 | 0.011761092 | 0.001019012 |
| Whisker Top (Max- Q3) | 0.006311378 | 0.005409753 | 0.00703068 | 0.003466637 |
| Whisker Bottom (Q1- Min) | 0.029151723 | 0.026883923 | 0.089478206 | 0.004308321 |

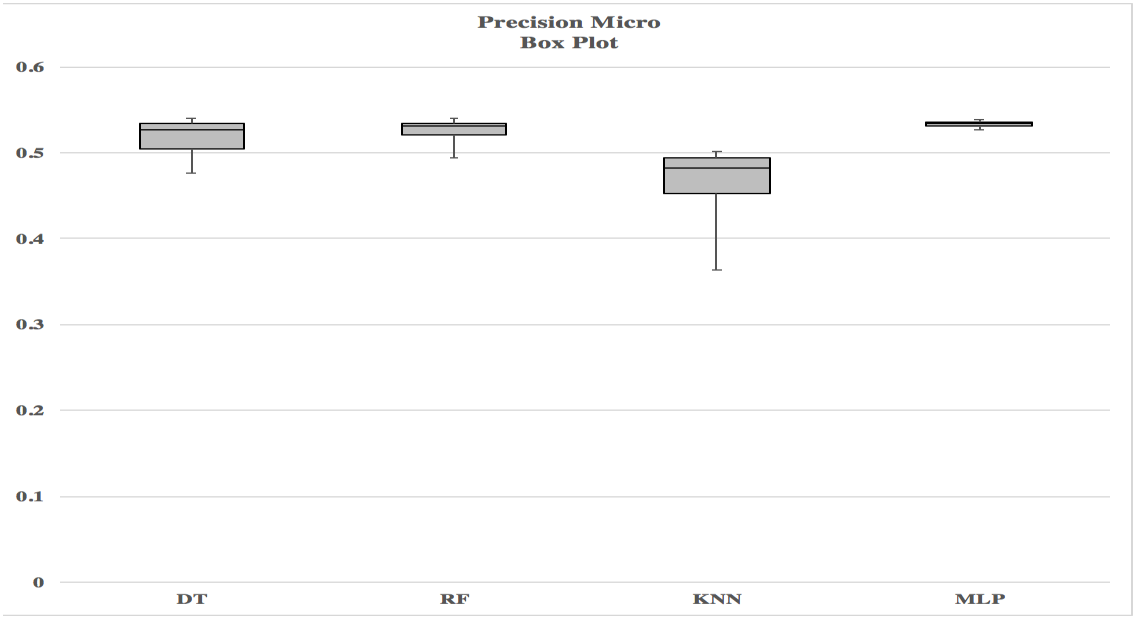


Figure 4.97. Precision Micro Box Plot for Readmission Dataset Using Suggested Features

Table 4.47. Precision Weighted Value for Readmission Dataset Using Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.291369365 | 0.291369365 | 0.434514215 | 0.408176626 |
| First Quartile (Q1) | 0.425381437 | 0.377573899 | 0.439425579 | 0.422266921 |
| Median Value | 0.426212772 | 0.426170801 | 0.441783 | 0.430038326 |
| Third Quartile(Q3) | 0.429365844 | 0.432154358 | 0.442745252 | 0.436828617 |
| Max Value | 0.456142085 | 0.460733371 | 0.444099773 | 0.442247236 |
| Box 1-hidden (Q1) | 0.425381437 | 0.377573899 | 0.439425579 | 0.422266921 |
| Box 2 (Median - Q1) | 0.000831334 | 0.048596902 | 0.002357421 | 0.007771405 |
| Box 3 (Q3- Median) | 0.003153073 | 0.005983557 | 0.000962252 | 0.006790292 |
| Whisker Top (Max- Q3) | 0.026776241 | 0.028579013 | 0.001354522 | 0.005418618 |
| Whisker Bottom (Q1- Min) | 0.134012072 | 0.086204533 | 0.004911363 | 0.014090295 |

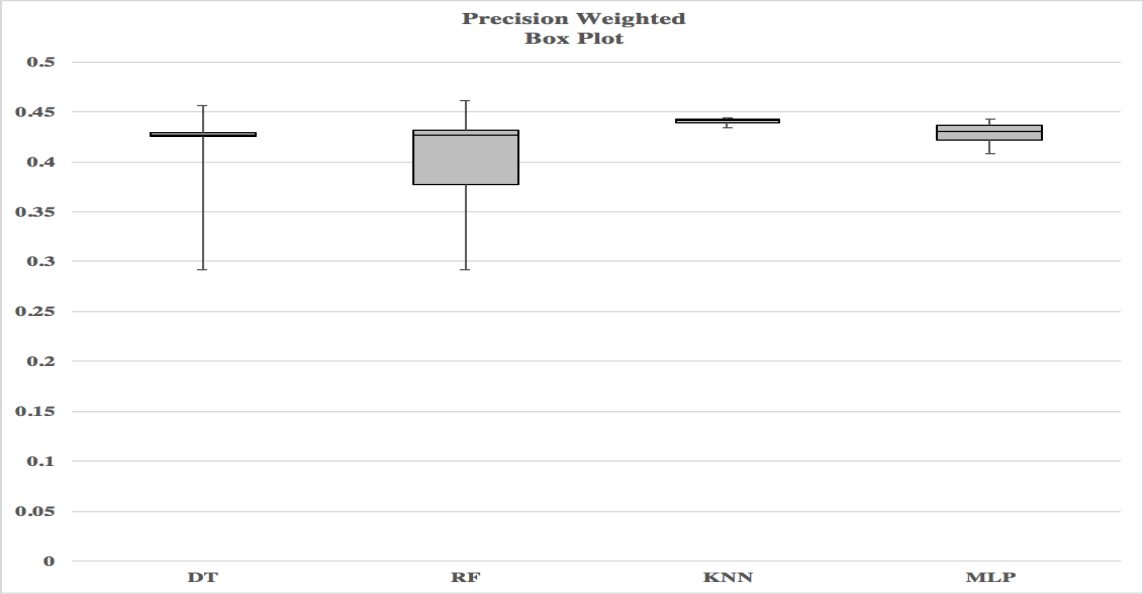


Figure 4.98. Precision Weighted Box Plot for Readmission Dataset Using Suggested Features

Table 4.48. Recall Macro Value for Readmission Dataset Using Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.330866895 | 0.331497333 | 0.341458532 | 0.332315095 |
| First Quartile (Q1) | 0.33233951 | 0.331838741 | 0.344507136 | 0.333664682 |
| Median Value | 0.332533356 | 0.332254772 | 0.344939058 | 0.336145794 |
| Third Quartile(Q3) | 0.333256244 | 0.332686403 | 0.345227326 | 0.337054968 |
| Max Value | 0.335560703 | 0.333334756 | 0.347103599 | 0.341938222 |
| Box 1-hidden (Q1) | 0.33233951 | 0.331838741 | 0.344507136 | 0.333664682 |
| Box 2 (Median - Q1) | 0.000193846 | 0.00041603 | 0.000431922 | 0.002481112 |
| Box 3 (Q3- Median) | 0.000722888 | 0.000431631 | 0.000288269 | 0.000909174 |
| Whisker Top (Max- Q3) | 0.002304459 | 0.000648352 | 0.001876273 | 0.004883254 |
| Whisker Bottom (Q1- Min) | 0.001472615 | 0.000341408 | 0.003048604 | 0.001349588 |

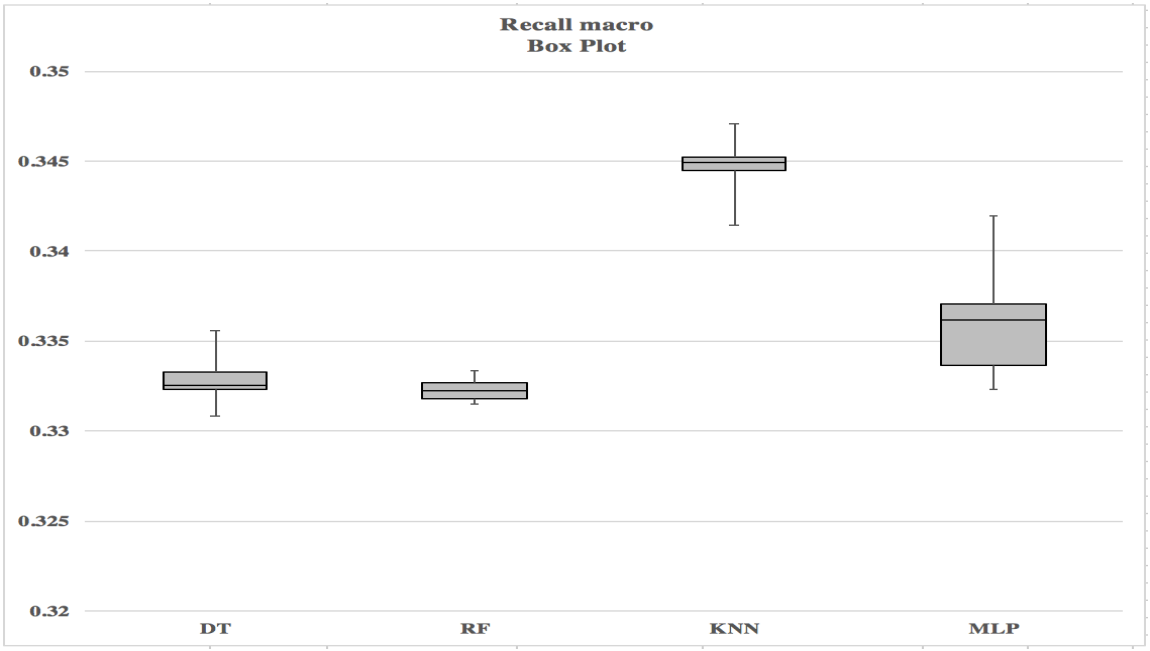


Figure 4.99. Recall Macro Box Plot for Readmission Dataset Using Suggested Features

Table 4.49. Recall Micro Value for Readmission Dataset Using Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.476467826 | 0.493860955 | 0.363447456 | 0.508416836 |
| First Quartile (Q1) | 0.504700441 | 0.520744877 | 0.452925662 | 0.527902934 |
| Median Value | 0.527623205 | 0.530944983 | 0.482861624 | 0.530095807 |
| Third Quartile(Q3) | 0.533475029 | 0.534376655 | 0.494622716 | 0.533884632 |
| Max Value | 0.539786407 | 0.539786407 | 0.501653396 | 0.537778355 |
| Box 1-hidden (Q1) | 0.504700441 | 0.520744877 | 0.452925662 | 0.527902934 |
| Box 2 (Median - Q1) | 0.022922765 | 0.010200106 | 0.029935962 | 0.002192873 |
| Box 3 (Q3- Median) | 0.005851824 | 0.003431671 | 0.011761092 | 0.003788825 |
| Whisker Top (Max- Q3) | 0.006311378 | 0.005409753 | 0.00703068 | 0.003893723 |
| Whisker Bottom (Q1- Min) | 0.028232614 | 0.026883923 | 0.089478206 | 0.019486098 |

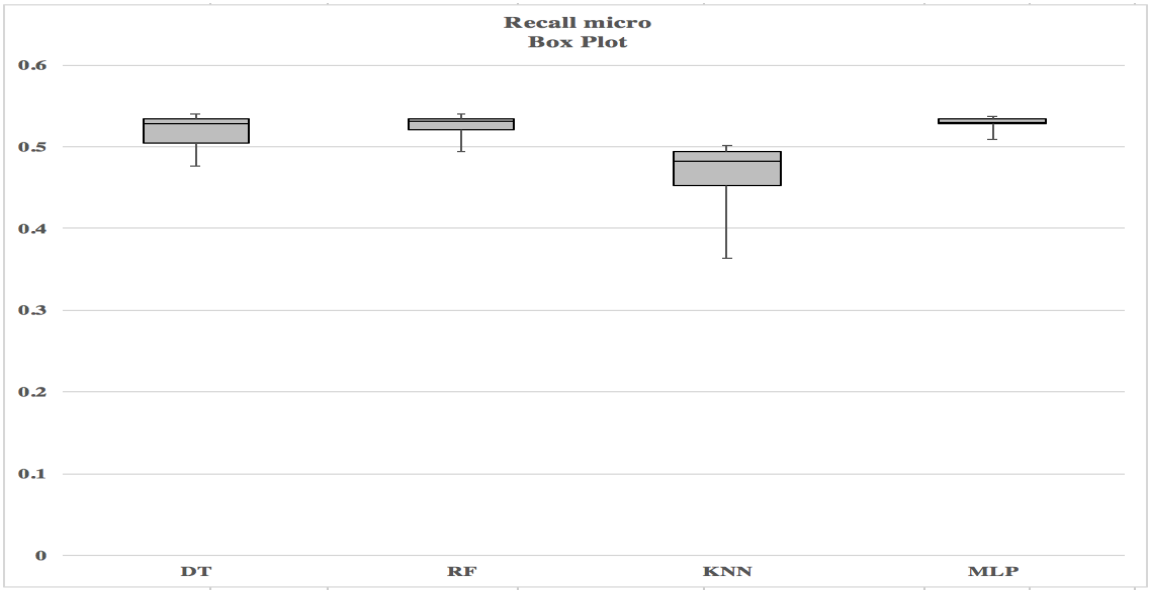


Figure 4.100. Recall Micro Box Plot for Readmission Dataset Using Suggested Features

Table 4.50. Recall Weighted Value for Readmission Dataset Using Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.476417875 | 0.493860955 | 0.363447456 | 0.516069413 |
| First Quartile (Q1) | 0.50507258 | 0.520744877 | 0.452925662 | 0.532830654 |
| Median Value | 0.527388433 | 0.530944983 | 0.482861624 | 0.53471133 |
| Third Quartile(Q3) | 0.533475029 | 0.534376655 | 0.494622716 | 0.536207379 |
| Max Value | 0.539786407 | 0.539786407 | 0.501653396 | 0.538867299 |
| Box 1-hidden (Q1) | 0.50507258 | 0.520744877 | 0.452925662 | 0.532830654 |
| Box 2 (Median - Q1) | 0.022315854 | 0.010200106 | 0.029935962 | 0.001880676 |
| Box 3 (Q3-Median) | 0.006086596 | 0.003431671 | 0.011761092 | 0.001496049 |
| Whisker Top (Max- Q3) | 0.006311378 | 0.005409753 | 0.00703068 | 0.00265992 |
| Whisker Bottom (Q1- Min) | 0.028654705 | 0.026883923 | 0.089478206 | 0.016761242 |

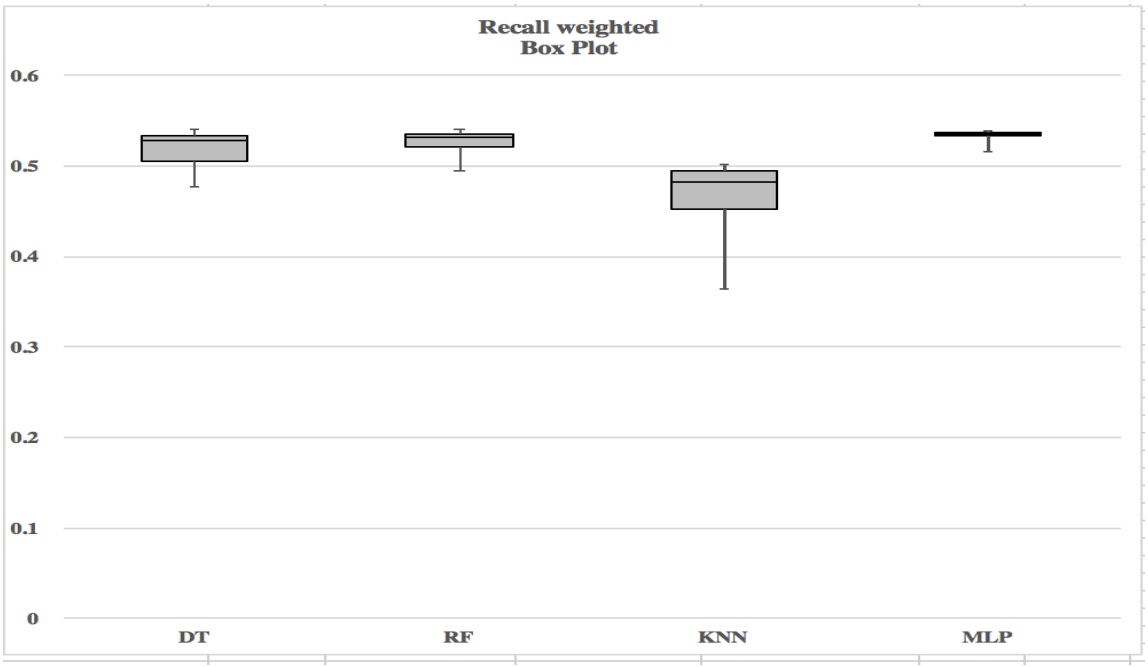


Figure 4.101. Recall Weighted Box Plot for Readmission Dataset Using Suggested Features

#### 4.5.3.3 Best Model

The following diagram shows the best model of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithms using suggested features.

| Algorithm | Accuracy | | Precision Micro | | Precision Macro | | Precision Weighted | | Recall Micro | | Recall Macro | | Recall Weighted | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value |
| Decission Tree | max_depth: 1 | 0.539786407 | max_depth: 1 | 0.539786407 | max_depth: 6 | 0.35966497 | max_depth: 4 | 0.456142085 | max_depth: 1 | 0.539786407 | max_depth: 20 | 0.335560703 | max_depth: 1 | 0.539786407 |
| Random Forest | max_depth: 1 | 0.539786407 | max_depth: 1 | 0.539786407 | max_depth: 10 | 0.407754946 | max_depth: 10 | 0.460733371 | max_depth: 1 | 0.539786407 | max_depth: 20 | 0.333334756 | max_depth: 1 | 0.539786407 |
| KNN | n_neighbors: 20 | 0.501653396 | n_neighbors: 20 | 0.501653396 | n_neighbors: 20 | 0.357126197 | n_neighbors: 20 | 0.444099773 | n_neighbors: 20 | 0.501653396 | n_neighbors: 5 | 0.347103599 | n_neighbors: 20 | 0.501653396 |
| MLP | max_iter: 17000 | 0.537698433 | max_iter: 60000 | 0.53859756 | max_iter: 10000 | 0.350645358 | max_iter: 60000 | 0.442247236 | max_iter: 170000 | 0.537778355 | max_iter: 100000 | 0.341938222 | max_iter: 160000 | 0.538867299 |

Figure 4.102. Best Model for Readmission Dataset Using Suggested Features

### 4.5.4 Using Transfer Learning Combined with Suggested Features

In this technique, top 10 important features identified during transfer learning are combined with all the expert suggested features as specified in section 3.3 were only used for training all the

models of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithms. The following sections shows the comparison between the models.

#### 4.5.4.1 Line Graph

Following line diagram shows the comparison of different models of each algorithm based on evaluation metrics.

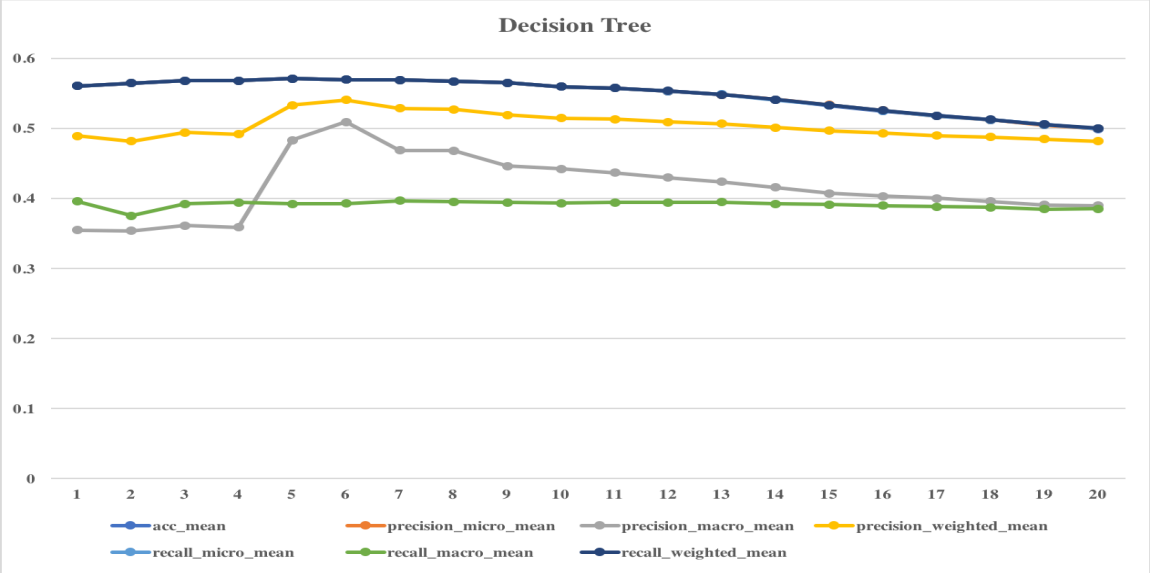


Figure 4.103. Line Graph of Decision Tree with Varying Max_Depth for Readmission Dataset Using Transfer Learning Combined with Suggested Features

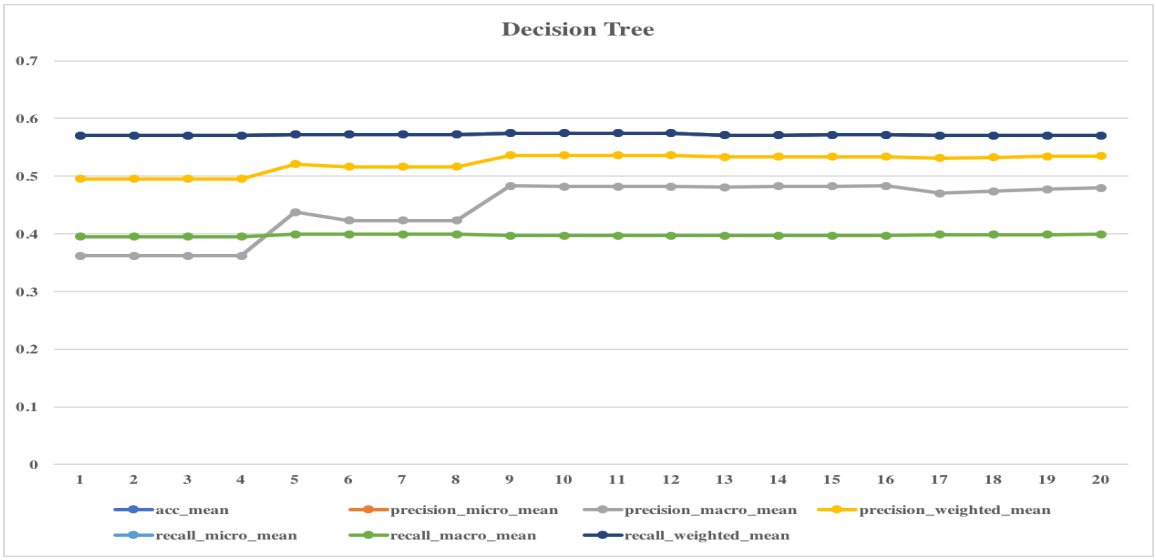


Figure 4.104. Line Graph of Decision Tree with Varying Max_Depth and Min_Sample_Split for Readmission Dataset Using Transfer Learning Combined with Suggested Features

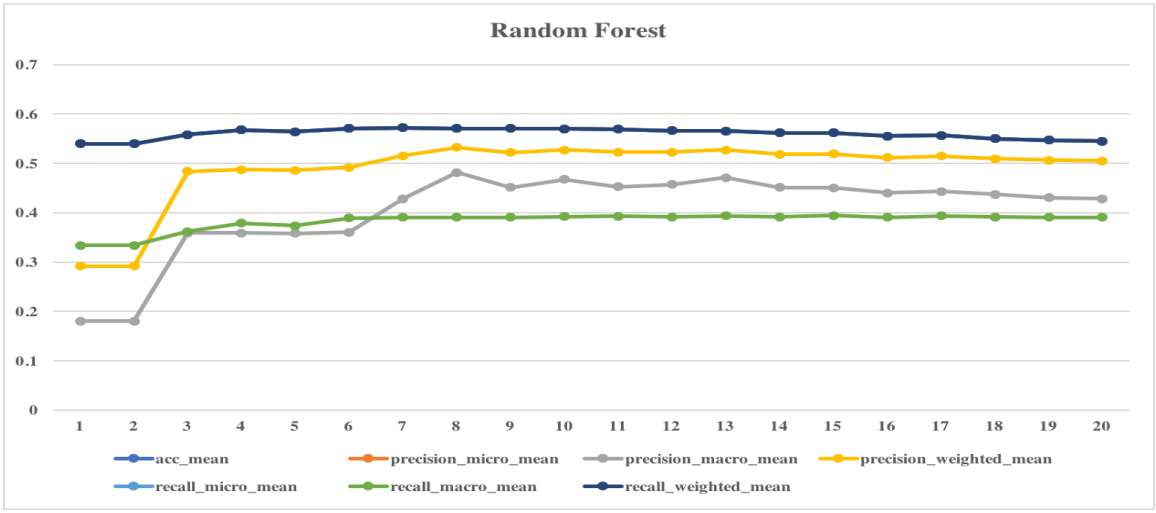


Figure 4.105. Line Graph of Random with Varying Max_Depth for Readmission Dataset Using Transfer Learning Combined with Suggested Features

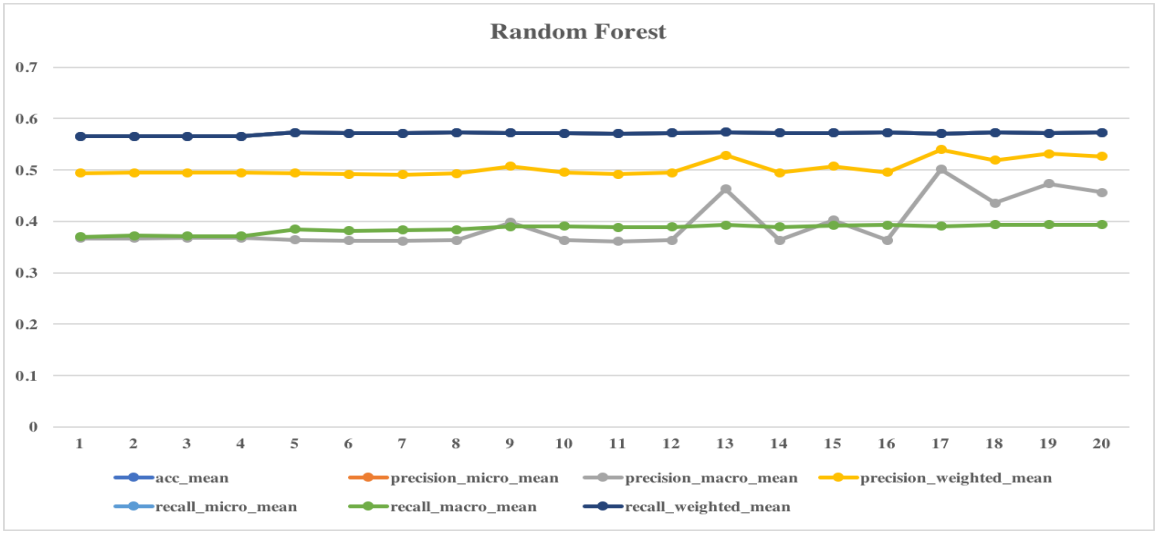


Figure 4.106. Line Graph of Random Forest with Varying Max_Depth and Min_Sample_Split for Readmission Dataset Using Transfer Learning Combined with Suggested Features

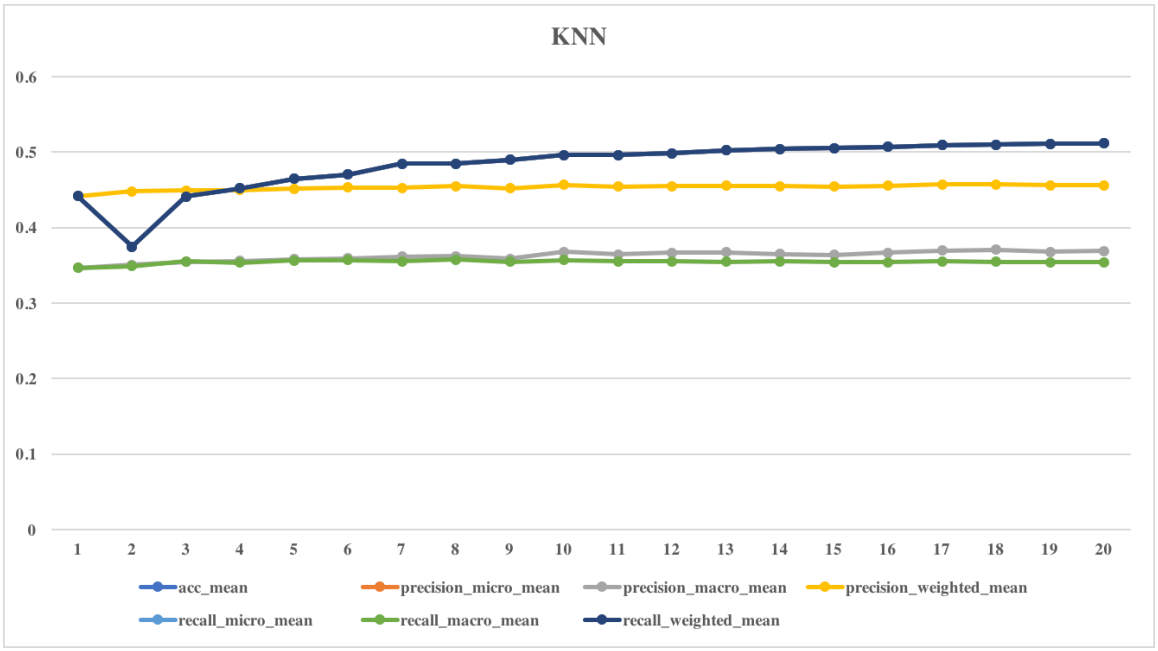


Figure 4.107. Line Graph of KNN with Varying N_Neighbor for Readmission Dataset Using Transfer Learning Combined with Suggested Features

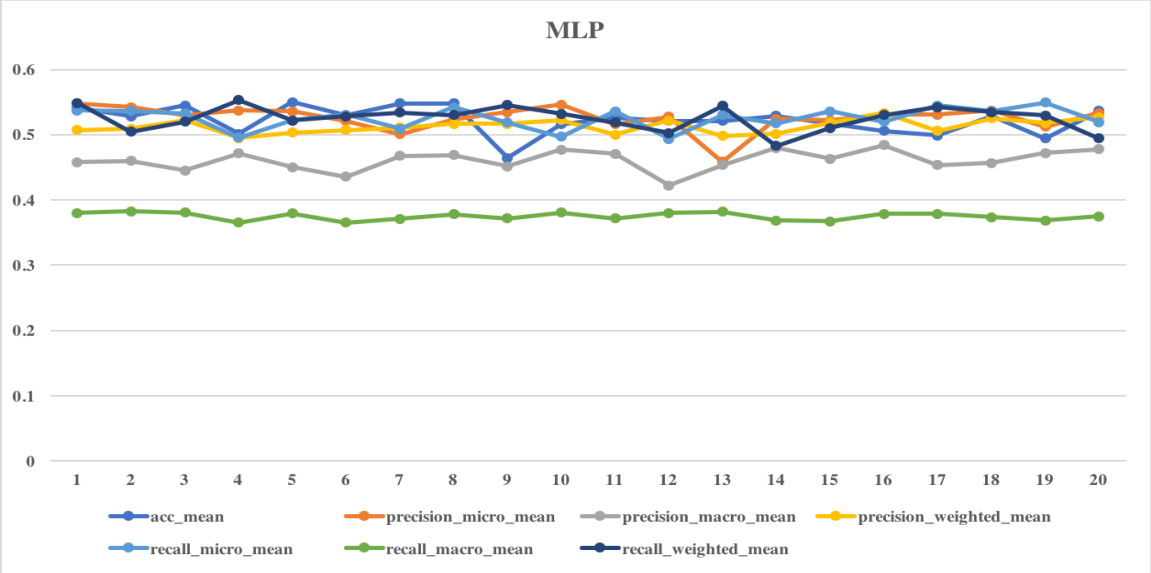


Figure 4.108. Line Graph of MLP with Varying Max_Iteration for Readmission Dataset Using Transfer Learning Combined with Suggested Features

#### 4.5.4.2 Box Plot

Following box diagram, shows the comparison of each algorithm based on grid search with 5-fold cross validation.

Table 4.51. Accuracy Value for Readmission Dataset Using Transfer Learning Combined with Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.499055916 | 0.539786407 | 0.375106147 | 0.464199726 |
| First Quartile (Q1) | 0.530482932 | 0.553797816 | 0.46893763 | 0.513619289 |
| Median Value | 0.558508247 | 0.563213683 | 0.496428464 | 0.527203612 |
| Third Quartile(Q3) | 0.567012498 | 0.569555032 | 0.505592076 | 0.537673457 |
| Max Value | 0.571026105 | 0.572564612 | 0.511983376 | 0.550086416 |
| Box 1-hidden (Q1) | 0.530482932 | 0.553797816 | 0.46893763 | 0.513619289 |
| Box 2 (Median - Q1) | 0.028025315 | 0.009415867 | 0.027490834 | 0.013584323 |
| Box 3 (Q3-Median) | 0.008504251 | 0.006341349 | 0.009163611 | 0.010469844 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Whisker Top (Max- Q3) | 0.004013607 | 0.003009581 | 0.0063913 | 0.012412959 |
| Whisker Bottom (Q1- Min) | 0.031427016 | 0.014011409 | 0.093831483 | 0.049419563 |

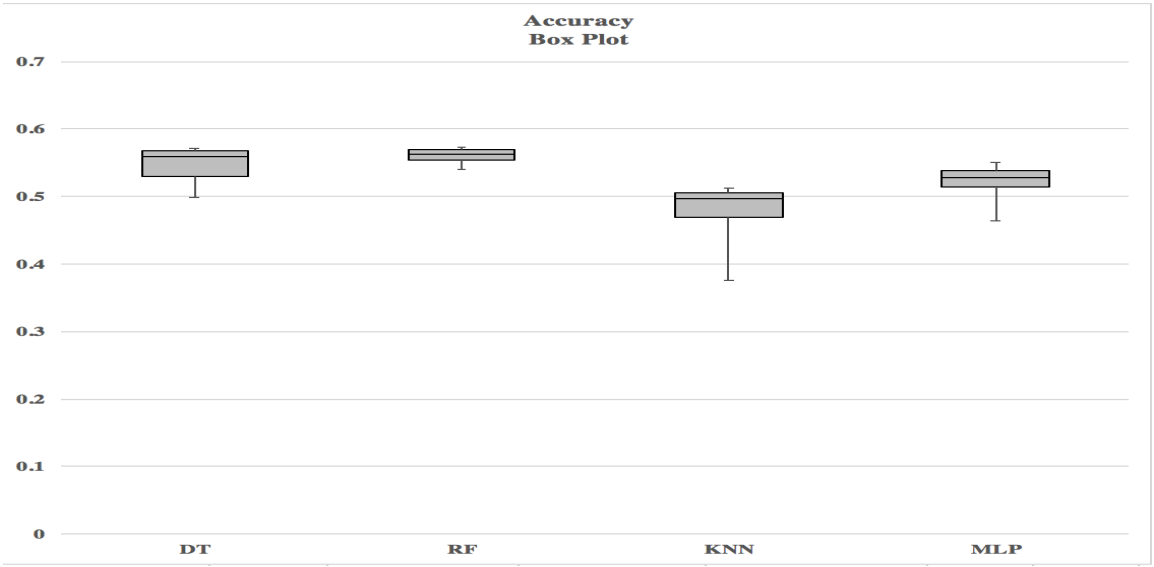


Figure 4.109. Accuracy Box Plot for Readmission Dataset Using Transfer Learning Combined with Suggested Features

Table 4.52. Precision Macro Value for Readmission Dataset Using Transfer Learning Combined with Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.353411859 | 0.179928802 | 0.346985021 | 0.422019359 |
| First Quartile (Q1) | 0.39191337 | 0.36007723 | 0.358659192 | 0.452797222 |
| Median Value | 0.412674852 | 0.438481329 | 0.364331625 | 0.461352284 |
| Third Quartile(Q3) | 0.443437898 | 0.451574215 | 0.367458477 | 0.471351092 |
| Max Value | 0.508752218 | 0.481678106 | 0.37094636 | 0.484072169 |
| Box 1-hidden (Q1) | 0.39191337 | 0.36007723 | 0.358659192 | 0.452797222 |
| Box 2 (Median - Q1) | 0.020761482 | 0.0784041 | 0.005672433 | 0.008555061 |
| Box 3 (Q3- Median) | 0.030763045 | 0.013092886 | 0.003126852 | 0.009998809 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Whisker Top (Max- Q3) | 0.065314321 | 0.030103891 | 0.003487883 | 0.012721076 |
| Whisker Bottom (Q1- Min) | 0.03850151 | 0.180148427 | 0.011674171 | 0.030777864 |

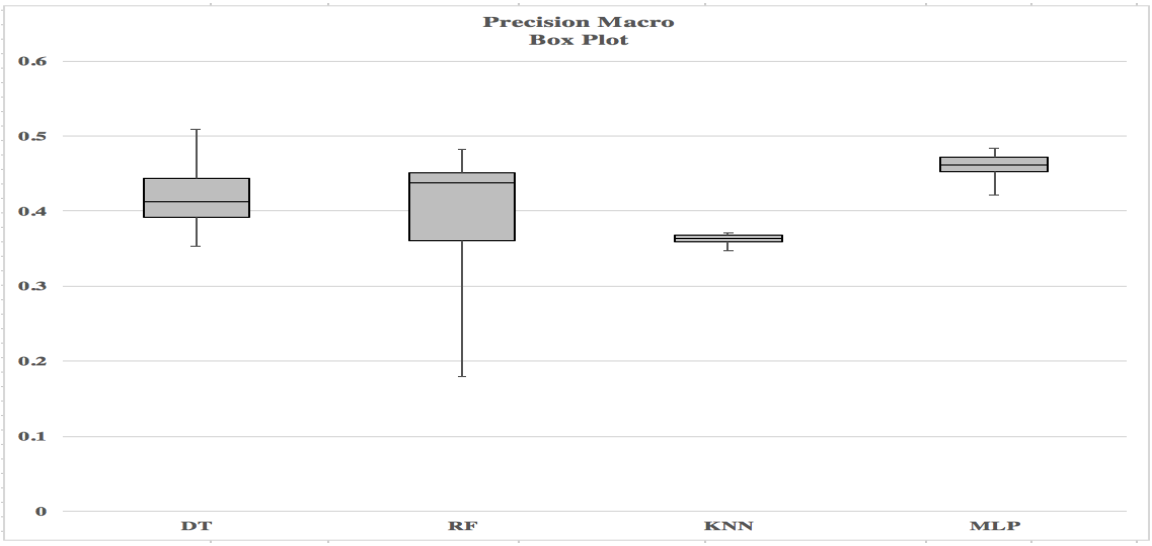


Figure 4.110. Precision Macro Box Plot for Readmission Dataset Using Transfer Learning Combined with Suggested Features.

Table 4.53. Precision Micro Value for Readmission Dataset Using Transfer Learning Combined with Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.498776187 | 0.539786407 | 0.375106147 | 0.458485269 |
| First Quartile (Q1) | 0.53109234 | 0.553797816 | 0.46893763 | 0.521836319 |
| Median Value | 0.55844331 | 0.563213683 | 0.496428464 | 0.530155749 |
| Third Quartile(Q3) | 0.566885121 | 0.569555032 | 0.505592076 | 0.535892684 |
| Max Value | 0.571026105 | 0.572564612 | 0.511983376 | 0.547838597 |
| Box 1-hidden (Q1) | 0.53109234 | 0.553797816 | 0.46893763 | 0.521836319 |
| Box 2 (Median - Q1) | 0.02735097 | 0.009415867 | 0.027490834 | 0.00831943 |
| Box 3 (Q3- Median) | 0.008441811 | 0.006341349 | 0.009163611 | 0.005736935 |
| Whisker Top (Max- Q3) | 0.004140983 | 0.003009581 | 0.0063913 | 0.011945912 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Whisker Bottom (Q1- Min) | 0.032316153 | 0.014011409 | 0.093831483 | 0.063351049 |

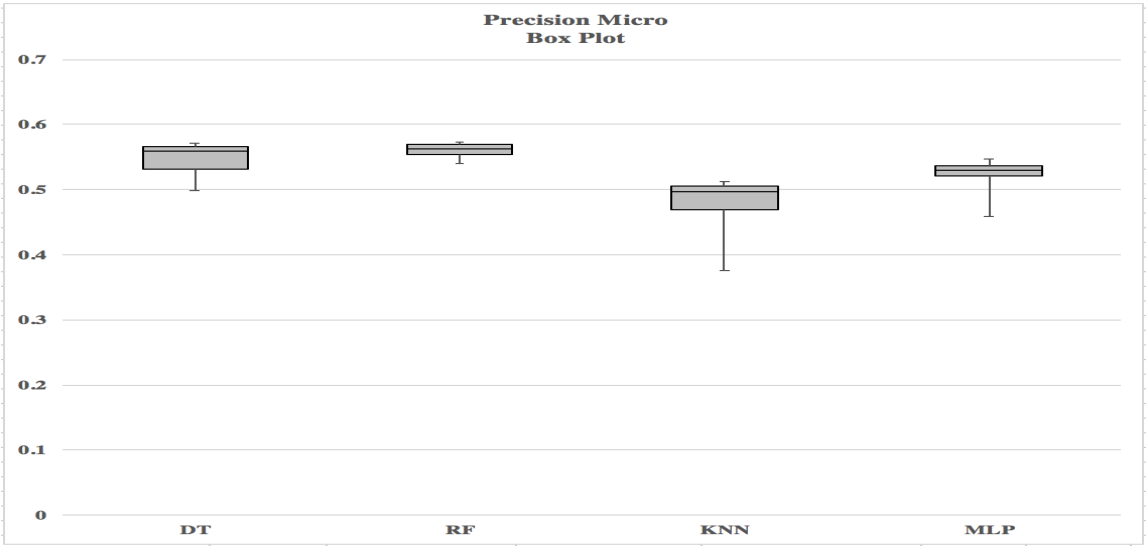


Figure 4.111. Precision Micro Box Plot for Readmission Dataset Using Transfer Learning Combined with Suggested Features

Table 4.54. Precision Weighted Value for Readmission Dataset Using Transfer Learning Combined with Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.481312591 | 0.291369365 | 0.441584148 | 0.494776808 |
| First Quartile (Q1) | 0.489788255 | 0.490456965 | 0.452127895 | 0.505238469 |
| Median Value | 0.498109111 | 0.513070354 | 0.454772809 | 0.513737591 |
| Third Quartile(Q3) | 0.515311245 | 0.522053363 | 0.455644785 | 0.52209574 |
| Max Value | 0.540476231 | 0.532518182 | 0.457267393 | 0.533138558 |
| Box 1-hidden (Q1) | 0.489788255 | 0.490456965 | 0.452127895 | 0.505238469 |
| Box 2 (Median - Q1) | 0.008320856 | 0.022613389 | 0.002644914 | 0.008499122 |
| Box 3 (Q3- Median) | 0.017202134 | 0.008983009 | 0.000871976 | 0.008358148 |
| Whisker Top (Max- Q3) | 0.025164986 | 0.010464819 | 0.001622608 | 0.011042819 |
| Whisker Bottom (Q1- Min) | 0.008475664 | 0.199087599 | 0.010543747 | 0.010461661 |

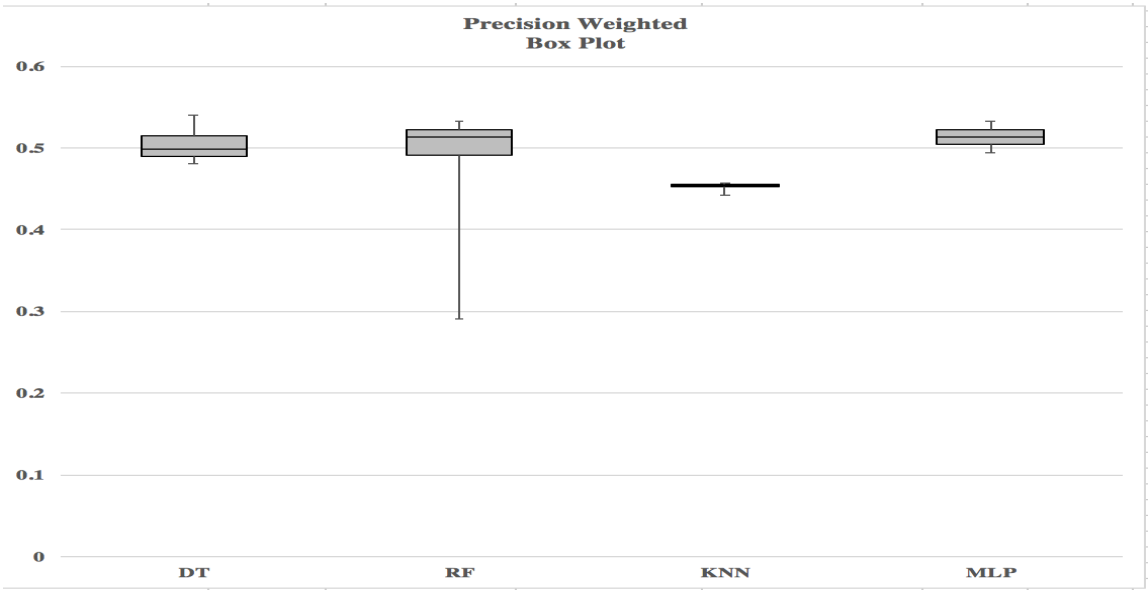


Figure 4.112. Precision Weighted Box Plot for Readmission Dataset Using Transfer Learning Combined with Suggested Features

Table 4.55. Recall Macro Value for Readmission Dataset Using Transfer Learning Combined with Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.374854727 | 0.333333333 | 0.346901452 | 0.365132534 |
| First Quartile (Q1) | 0.389217446 | 0.386354033 | 0.354197603 | 0.370204918 |
| Median Value | 0.392457809 | 0.390722433 | 0.355126243 | 0.376467014 |
| Third Quartile(Q3) | 0.394150168 | 0.391421398 | 0.355540295 | 0.379696771 |
| Max Value | 0.396432786 | 0.393934783 | 0.357606892 | 0.382357058 |
| Box 1-hidden (Q1) | 0.389217446 | 0.386354033 | 0.354197603 | 0.370204918 |
| Box 2 (Median - Q1) | 0.003240364 | 0.0043684 | 0.00092864 | 0.006262096 |
| Box 3 (Q3- Median) | 0.001692358 | 0.000698965 | 0.000414052 | 0.003229756 |
| Whisker Top (Max- Q3) | 0.002282618 | 0.002513385 | 0.002066596 | 0.002660288 |
| Whisker Bottom (Q1- Min) | 0.014362719 | 0.0530207 | 0.007296151 | 0.005072383 |

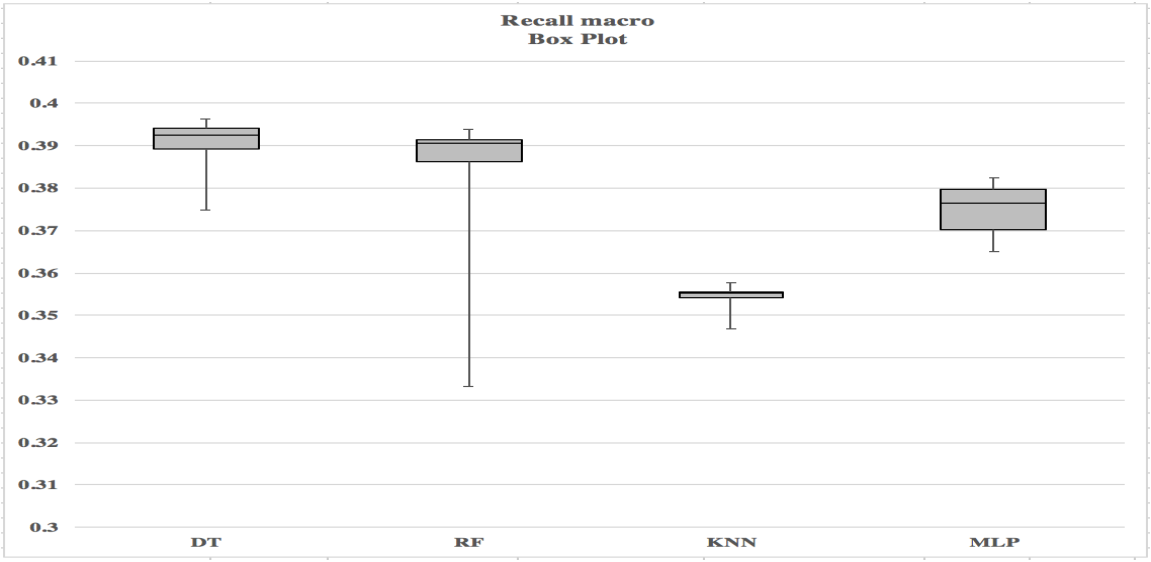


Figure 4.113. Recall Macro Box Plot for Readmission Dataset Using Transfer Learning Combined with Suggested Features

Table 4.56. Recall Micro Value for Readmission Dataset Using Transfer Learning Combined with Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.49885611 | 0.539786407 | 0.375106147 | 0.4937111 |
| First Quartile (Q1) | 0.530700221 | 0.553797816 | 0.46893763 | 0.518367184 |
| Median Value | 0.558458295 | 0.563213683 | 0.496428464 | 0.530520395 |
| Third Quartile(Q3) | 0.567034976 | 0.569555032 | 0.505592076 | 0.536142442 |
| Max Value | 0.571026105 | 0.572564612 | 0.511983376 | 0.549027443 |
| Box 1-hidden (Q1) | 0.530700221 | 0.553797816 | 0.46893763 | 0.518367184 |
| Box 2 (Median - Q1) | 0.027758075 | 0.009415867 | 0.027490834 | 0.012153211 |
| Box 3 (Q3- Median) | 0.008576681 | 0.006341349 | 0.009163611 | 0.005622047 |
| Whisker Top (Max- Q3) | 0.003991129 | 0.003009581 | 0.0063913 | 0.012885002 |
| Whisker Bottom (Q1- Min) | 0.031844111 | 0.014011409 | 0.093831483 | 0.024656084 |

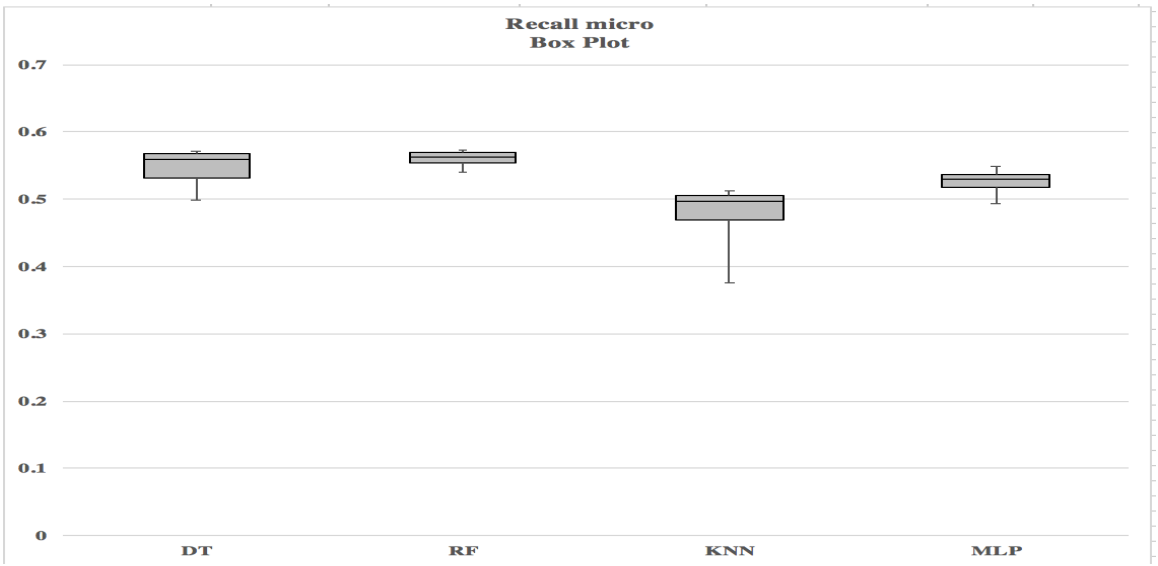


Figure 4.114. Recall Micro Box Plot for Readmission Dataset Using Transfer Learning Combined with Suggested Features

Table 4.57. Recall Weighted Value for Readmission Dataset Using Transfer Learning Combined with Suggested Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.499465518 | 0.539786407 | 0.375106147 | 0.482781702 |
| First Quartile (Q1) | 0.530637781 | 0.553797816 | 0.46893763 | 0.51655644 |
| Median Value | 0.558458295 | 0.563213683 | 0.496428464 | 0.529481403 |
| Third Quartile(Q3) | 0.566922585 | 0.569555032 | 0.505592076 | 0.536374717 |
| Max Value | 0.571026105 | 0.572564612 | 0.511983376 | 0.553333267 |
| Box 1-hidden (Q1) | 0.530637781 | 0.553797816 | 0.46893763 | 0.51655644 |
| Box 2 (Median - Q1) | 0.027820514 | 0.009415867 | 0.027490834 | 0.012924963 |
| Box 3 (Q3- Median) | 0.00846429 | 0.006341349 | 0.009163611 | 0.006893313 |
| Whisker Top (Max- Q3) | 0.00410352 | 0.003009581 | 0.0063913 | 0.01695855 |
| Whisker Bottom (Q1- Min) | 0.031172263 | 0.014011409 | 0.093831483 | 0.033774739 |

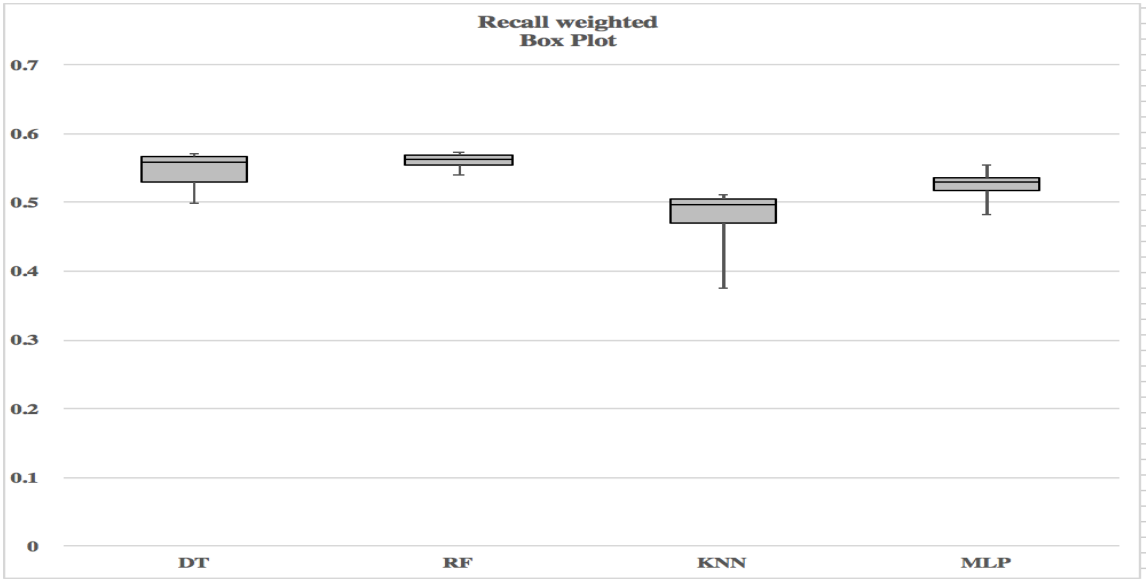


Figure 4.115. Recall Weighted Box Plot for Readmission Dataset Using Transfer Learning Combined with Suggested Features

### 4.5.4.3 Best Model

The following diagram shows the best model created varying only one coefficient of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithms using transfer learning combined with suggested features.

| Algorithm | Accuracy | | Precision Micro | | Precision Macro | | Precision Weighted | | Recall Micro | | Recall Macro | | Recall Weighted | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value |
| Decission Tree | max_depth: 5 | 0.571026105 | max_depth: 5 | 0.571026105 | max_depth: 6 | 0.508752218 | max_depth: 6 | 0.540476231 | max_depth: 5 | 0.571026105 | max_depth: 7 | 0.396432786 | max_depth: 5 | 0.571026105 |
| Random Forest | max_depth: 7 | 0.572564612 | max_depth: 7 | 0.572564612 | max_depth: 8 | 0.481678106 | max_depth: 8 | 0.532518182 | max_depth: 7 | 0.572564612 | max_depth: 15 | 0.393934783 | max_depth: 7 | 0.572564612 |
| KNN | n_neighbors: 20 | 0.511983376 | n_neighbors: 20 | 0.511983376 | n_neighbors: 18 | 0.37094636 | n_neighbors: 18 | 0.457267393 | n_neighbors: 20 | 0.511983376 | n_neighbors: 8 | 0.357606892 | n_neighbors: 20 | 0.511983376 |
| MLP | max_iter: 50000 | 0.550086416 | max_iter: 10000 | 0.547838597 | max_iter: 160000 | 0.484072169 | max_iter: 160000 | 0.533138558 | max_iter: 190000 | 0.549027443 | max_iter: 20000 | 0.382357058 | max_iter: 40000 | 0.553333267 |

Figure 4.116. Best Model for Readmission Dataset Using Transfer Learning Combined with Suggested Features.

### 4.5.5 Methodology and Algorithm Comparison Based on Accuracy for Readmission Dataset

In this section, we compare all the 4-methodologies used with readmission dataset, following table shows the comparisons between best models for each methodology and each machine learning algorithms.

Table 4.58. Accuracy Based Comparisons of Best Model for Readmission Dataset

| Features Used for Training | Best Model Accuracy with Grid Search Evaluation | | | |
|---|---|---|---|---|
| | Decision Tree | Random Forest | KNN | MLP |
| **All 45 features** | 0.573843371929 | 0.567979060311 | 0.508436816288 | 0.553912704676 |
| **Transfer learning with top 10 features** | 0.573733478526 | 0.575461802052 | 0.512183182 | 0.554671968191 |
| **With expert suggested features** | 0.539786407185 | 0.539786407185 | 0.501653396206 | 0.53769843252 |
| **Transfer learning combined with suggested features** | 0.571026104678 | 0.572564612326 | 0.511983376125 | 0.550086416176 |

The table 9 shows, the transfer learning methodology has better or almost same accuracy for all the machine learning algorithms comparing to other methodology. Here it also shows that the Decision Tree algorithm outperformed to be best among all the machine learning algorithm for all the methodology. It also shows that the model trained with only expert suggested features has the lowest accuracy for each algorithm.

## 4.6 BMI Dataset

We did different experiments on the BMI dataset by considering the machine learning algorithms and transfer learning technique. In this dataset, we have 73781 number of patient records with 118 different number of features.

## 4.7 Result and Finding with BMI Dataset

In this section, we mention detail results obtained for BMI dataset by following methodologies specified in chapter 3.

### 4.7.1 Using All the Features of Dataset

In this technique, we created the different models of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithms using all the feature of the BMI dataset. And each trained model performance is estimated by calculation evaluation metrics using grid search with 5-fold cross validation. We also compared the different models of the different algorithm as shown below.

#### 4.7.1.1 Line Graph

Following line diagram shows the comparison of different models of each algorithm based on evaluation metrics.

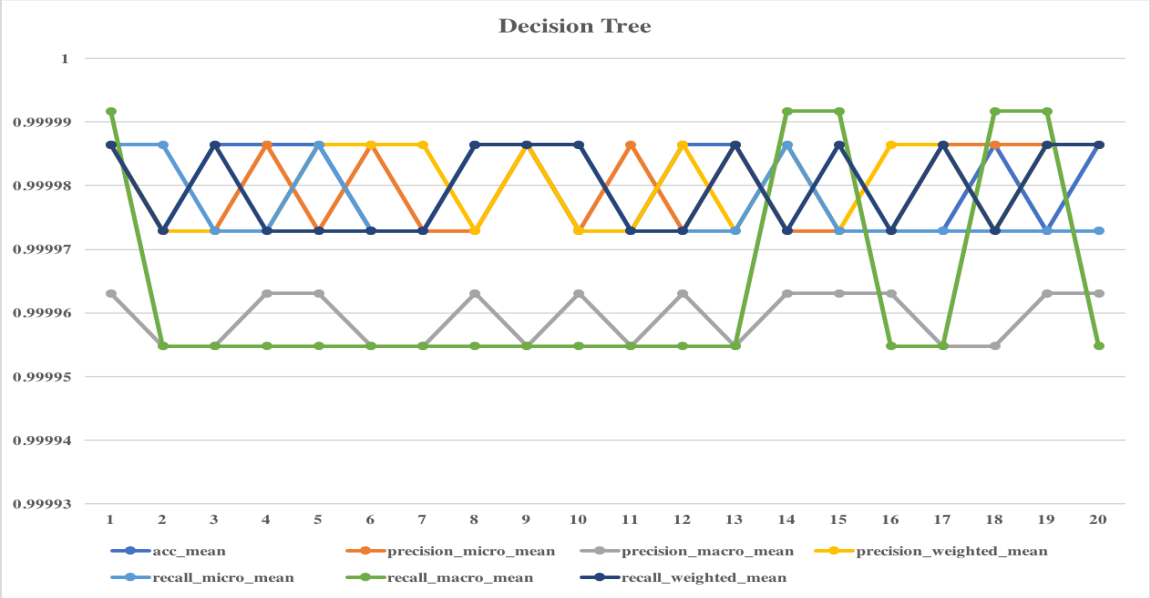


Figure 4.117. Line Graph of Decision Tree with Varying Max_Depth for BMI Dataset Using All Features

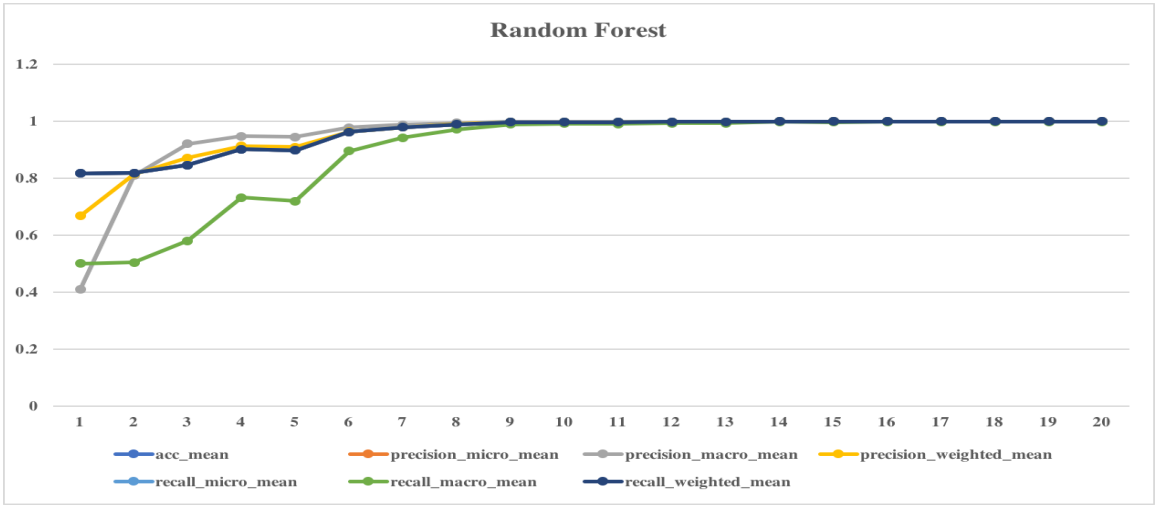


Figure 4.118. Line Graph of Random Forest with Varying Max_Depth for BMI Dataset Using All Features

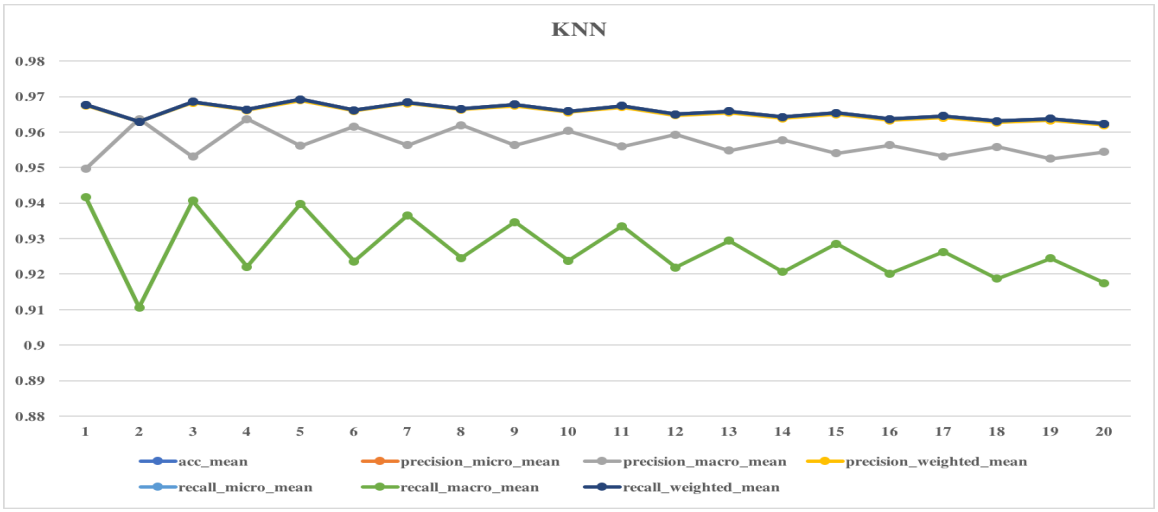


Figure 4.119. Line Graph of KNN with Varying N_Neighbor for BMI Dataset Using All Features

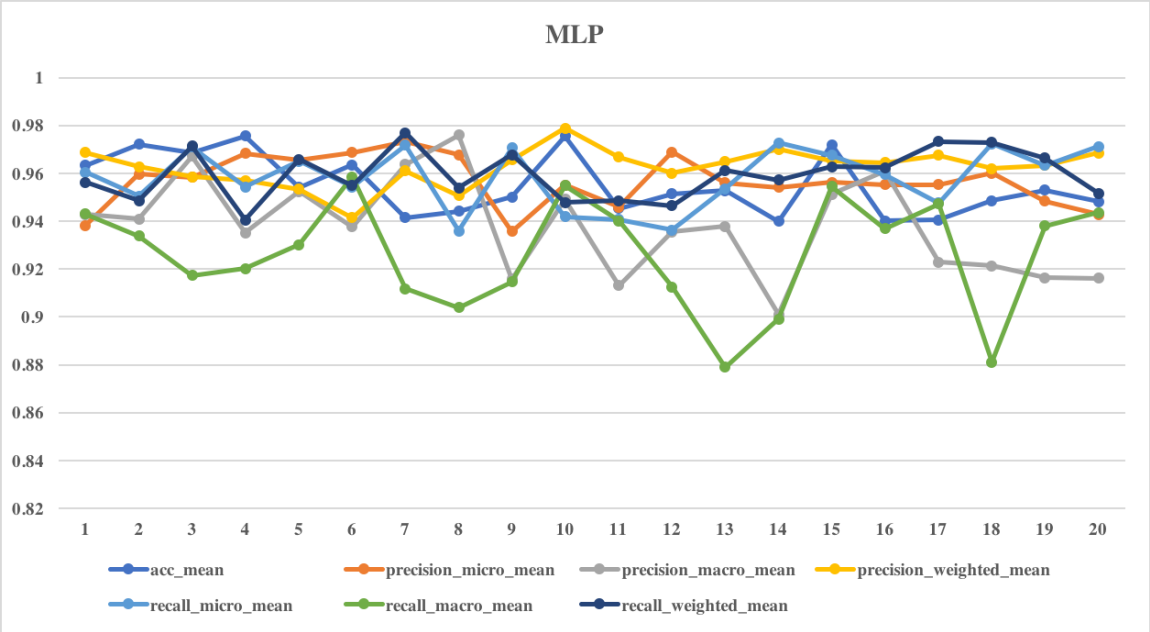


Figure 4.120. Line Graph of MLP with Varying Max_Iteration for BMI Dataset Using All Features

#### 4.7.1.2 Box Plot

Following box diagram, shows the comparison of each algorithm based on grid search with 5-fold cross validation.

Table 4.59. Accuracy Value for BMI Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.999972893 | 0.816538133 | 0.96234803 | 0.939916781 |
| First Quartile (Q1) | 0.999972893 | 0.946449628 | 0.964130332 | 0.944965506 |
| Median Value | 0.99997967 | 0.996204985 | 0.965878749 | 0.952087936 |
| Third Quartile(Q3) | 0.999986446 | 0.998759843 | 0.967457747 | 0.964794459 |
| Max Value | 0.999986446 | 0.999119014 | 0.969206164 | 0.97565769 |
| Box 1-hidden (Q1) | 0.999972893 | 0.946449628 | 0.964130332 | 0.944965506 |
| Box 2 (Median - Q1) | 6.77681E-06 | 0.049755357 | 0.001748418 | 0.00712243 |
| Box 3 (Q3-Median) | 6.77681E-06 | 0.002554858 | 0.001578997 | 0.012706523 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Whisker Top (Max- Q3) | 0 | 0.000359171 | 0.001748418 | 0.01086323 |
| Whisker Bottom (Q1- Min) | 0 | 0.129911495 | 0.001782302 | 0.005048725 |

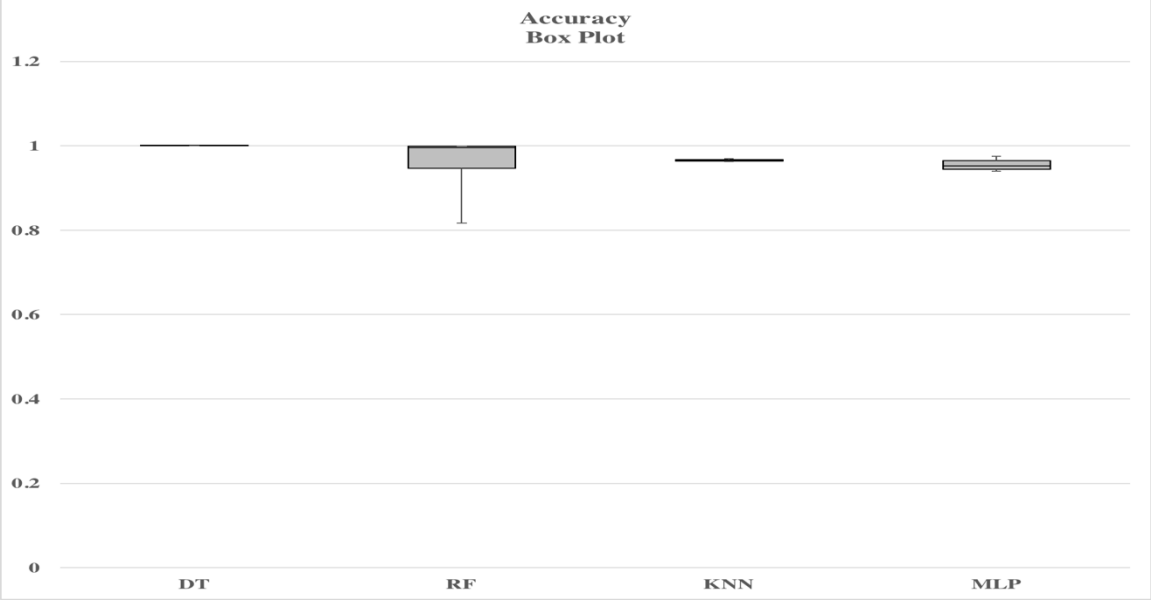


Figure 4.121. Accuracy Box Plot for BMI Dataset Using All Features

Table 4.60. Precision Macro Value for BMI Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.999954774 | 0.408269067 | 0.949663564 | 0.900990753 |
| First Quartile (Q1) | 0.999954774 | 0.96973092 | 0.954277554 | 0.920054855 |
| Median Value | 0.999963073 | 0.997216703 | 0.956205426 | 0.937776516 |
| Third Quartile(Q3) | 0.999963073 | 0.998622671 | 0.959542818 | 0.951542546 |
| Max Value | 0.999963073 | 0.999206103 | 0.963638339 | 0.975958119 |
| Box 1-hidden (Q1) | 0.999954774 | 0.96973092 | 0.954277554 | 0.920054855 |
| Box 2 (Median - Q1) | 8.29921E-06 | 0.027485783 | 0.001927872 | 0.017721661 |
| Box 3 (Q3- Median) | 0 | 0.001405968 | 0.003337392 | 0.01376603 |
| Whisker Top (Max- Q3) | 0 | 0.000583432 | 0.004095521 | 0.024415573 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Whisker Bottom (Q1- Min) | 0 | 0.561461854 | 0.004613991 | 0.019064103 |

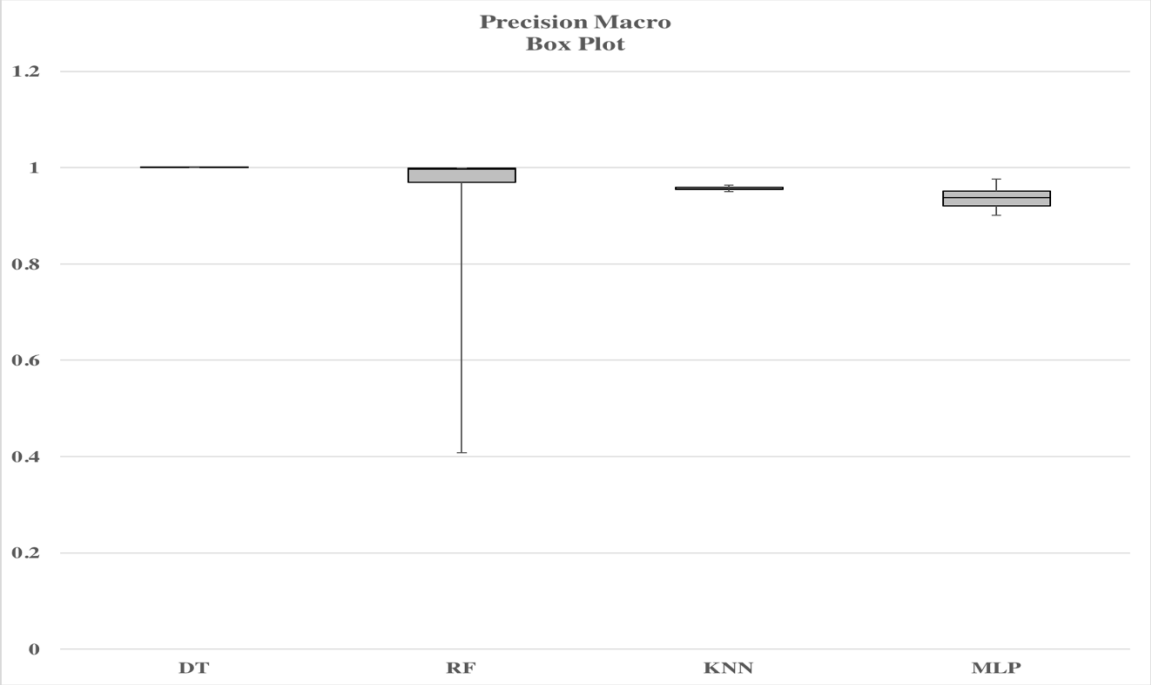


Figure 4.122. Precision Macro Box Plot for BMI Dataset Using All Features

Table 4.61. Precision Micro Value for BMI Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.999972893 | 0.816538133 | 0.96234803 | 0.935728711 |
| First Quartile (Q1) | 0.999972893 | 0.946449628 | 0.964130332 | 0.952602974 |
| Median Value | 0.99997967 | 0.996204985 | 0.965878749 | 0.956126916 |
| Third Quartile(Q3) | 0.999986446 | 0.998759843 | 0.967457747 | 0.966021062 |
| Max Value | 0.999986446 | 0.999119014 | 0.969206164 | 0.973001179 |
| Box 1-hidden (Q1) | 0.999972893 | 0.946449628 | 0.964130332 | 0.952602974 |
| Box 2 (Median - Q1) | 6.77681E-06 | 0.049755357 | 0.001748418 | 0.003523942 |
| Box 3 (Q3-Median) | 6.77681E-06 | 0.002554858 | 0.001578997 | 0.009894146 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Whisker Top (Max- Q3) | 0 | 0.000359171 | 0.001748418 | 0.006980117 |
| Whisker Bottom (Q1- Min) | 0 | 0.129911495 | 0.001782302 | 0.016874263 |

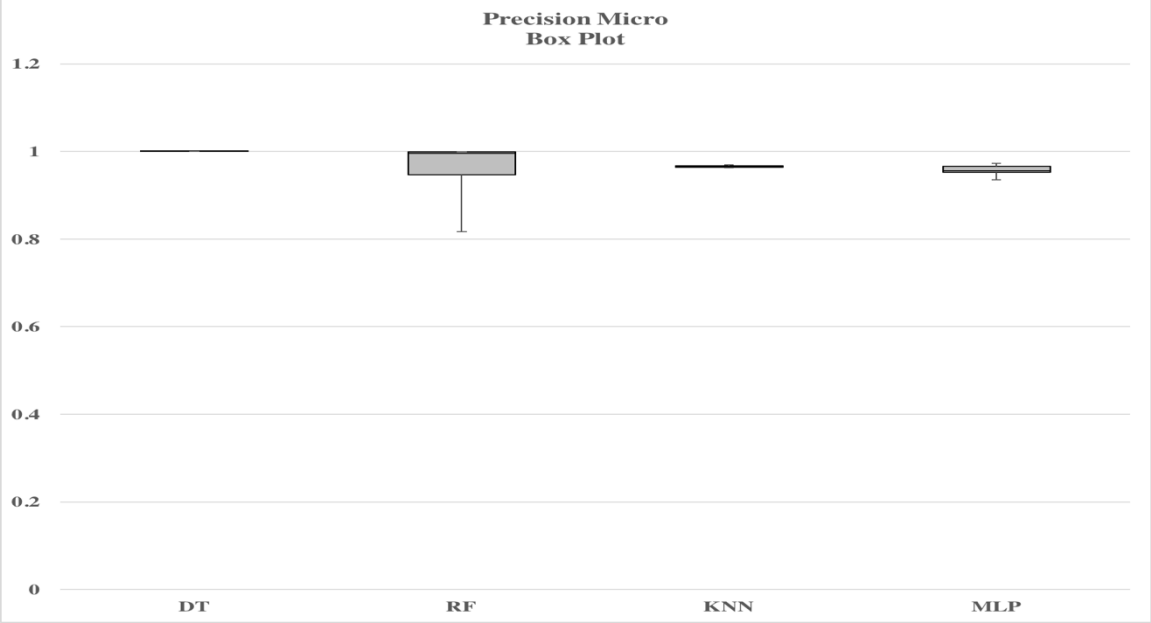


Figure 4.123. Precision Micro Box Plot for BMI Dataset Using All Features

Table 4.62. Precision Weighted Value for BMI Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.999972899 | 0.666734523 | 0.961911581 | 0.941472628 |
| First Quartile (Q1) | 0.999972899 | 0.950833312 | 0.963750706 | 0.959671722 |
| Median Value | 0.999986451 | 0.996223931 | 0.965516946 | 0.96390086 |
| Third Quartile(Q3) | 0.999986451 | 0.998759676 | 0.967093732 | 0.966892176 |
| Max Value | 0.999986451 | 0.999119204 | 0.968880655 | 0.978893515 |
| Box 1-hidden (Q1) | 0.999972899 | 0.950833312 | 0.963750706 | 0.959671722 |
| Box 2 (Median - Q1) | 1.35525E-05 | 0.045390619 | 0.00176624 | 0.004229138 |
| Box 3 (Q3- Median) | 0 | 0.002535744 | 0.001576787 | 0.002991316 |
| Whisker Top (Max- Q3) | 0 | 0.000359529 | 0.001786923 | 0.012001339 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Whisker Bottom (Q1- Min) | 0 | 0.284098789 | 0.001839125 | 0.018199094 |

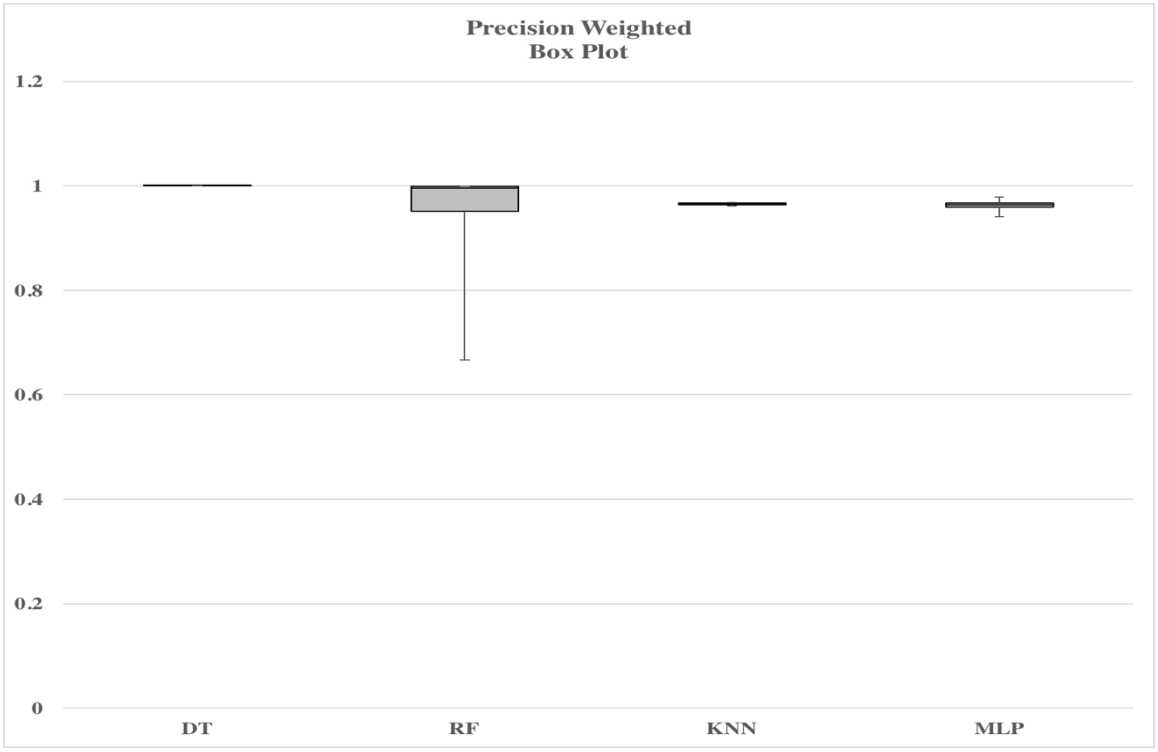


Figure 4.124. Precision Weighted Box Plot for BMI Dataset Using All Features

Table 4.63. Recall Macro Value for BMI Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.999954771 | 0.5 | 0.910567284 | 0.879043129 |
| First Quartile (Q1) | 0.999954771 | 0.854182959 | 0.921553775 | 0.912293407 |
| Median Value | 0.999954771 | 0.990116295 | 0.924489112 | 0.93194104 |
| Third Quartile(Q3) | 0.999964003 | 0.997235822 | 0.933772824 | 0.943175786 |
| Max Value | 0.999991701 | 0.997856778 | 0.941731289 | 0.958475168 |
| Box 1-hidden (Q1) | 0.999954771 | 0.854182959 | 0.921553775 | 0.912293407 |
| Box 2 (Median - Q1) | 0 | 0.135933336 | 0.002935337 | 0.019647633 |
| Box 3 (Q3-Median) | 9.23241E-06 | 0.007119527 | 0.009283712 | 0.011234746 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Whisker Top (Max- Q3) | 2.76972E-05 | 0.000620956 | 0.007958465 | 0.015299382 |
| Whisker Bottom (Q1- Min) | 0 | 0.354182959 | 0.010986491 | 0.033250278 |

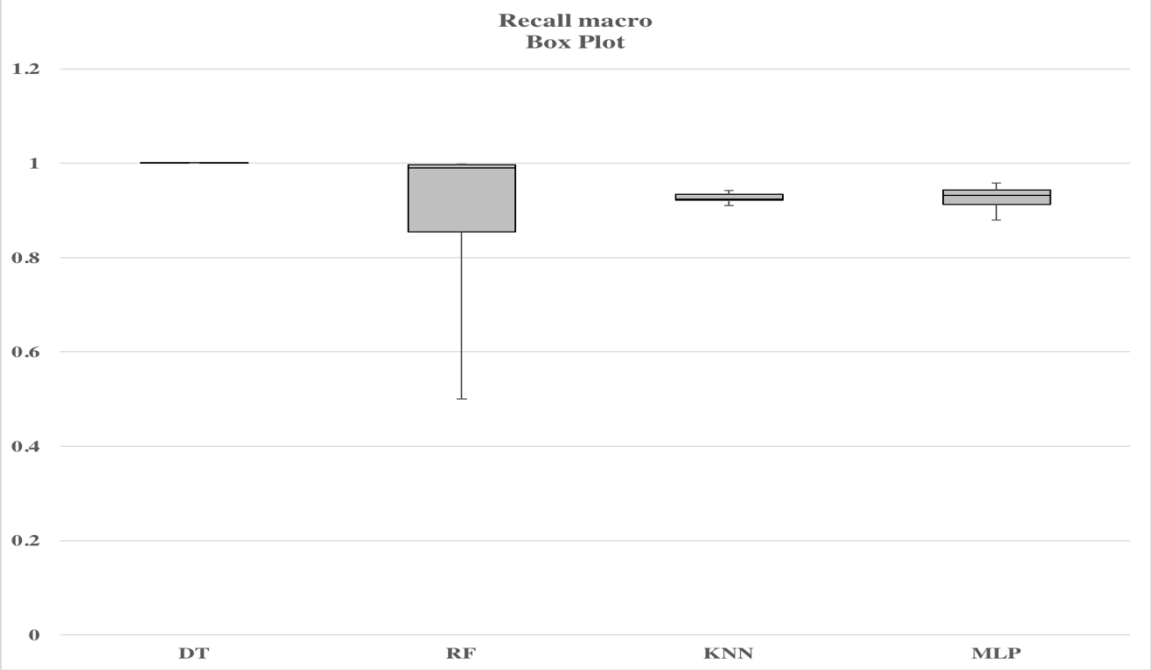


Figure 4.125. Recall Macro Box Plot for BMI Dataset Using All Features

Table 4.64. Recall Micro Value for BMI Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.999972893 | 0.816538133 | 0.96234803 | 0.93598623 |
| First Quartile (Q1) | 0.999972893 | 0.946449628 | 0.964130332 | 0.949716052 |
| Median Value | 0.999972893 | 0.996204985 | 0.965878749 | 0.959962592 |
| Third Quartile(Q3) | 0.999986446 | 0.998759843 | 0.967457747 | 0.97070384 |
| Max Value | 0.999986446 | 0.999119014 | 0.969206164 | 0.972621678 |
| Box 1-hidden (Q1) | 0.999972893 | 0.946449628 | 0.964130332 | 0.949716052 |
| Box 2 (Median - Q1) | 0 | 0.049755357 | 0.001748418 | 0.01024654 |
| Box 3 (Q3- Median) | 1.35536E-05 | 0.002554858 | 0.001578997 | 0.010741248 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Whisker Top (Max- Q3) | 0 | 0.000359171 | 0.001748418 | 0.001917838 |
| Whisker Bottom (Q1- Min) | 0 | 0.129911495 | 0.001782302 | 0.013729822 |

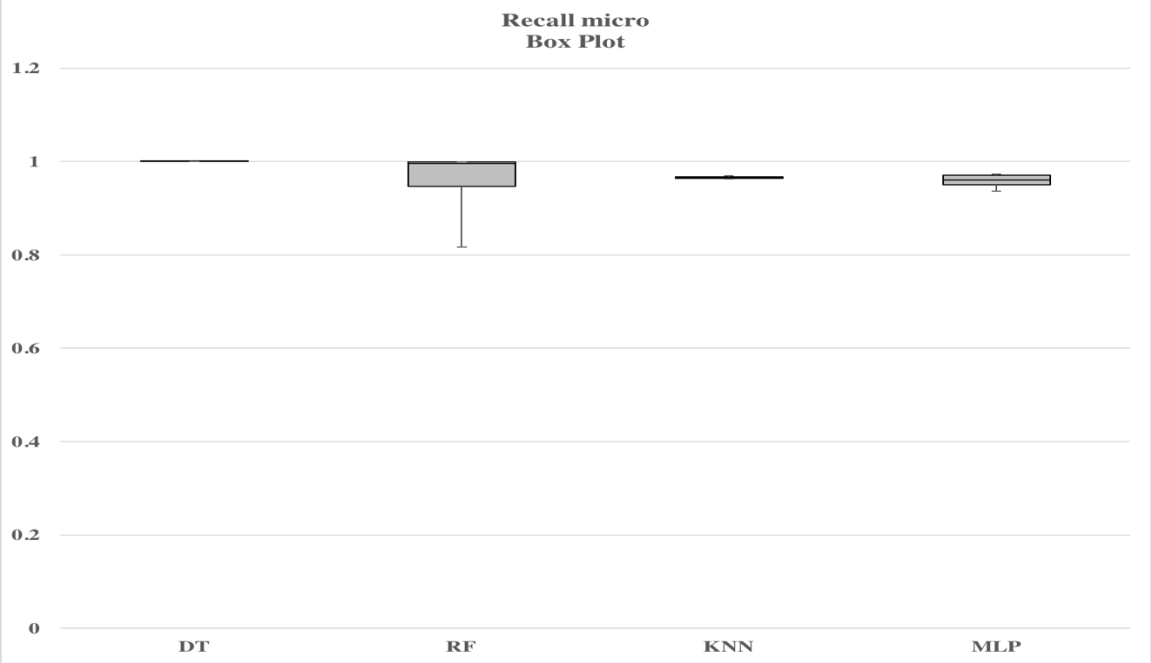


Figure 4.126. Recall Micro Box Plot for BMI Dataset Using All Features

Table 4.65. Recall Weighted Value for BMI Dataset Using All Features

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.999972893 | 0.816538133 | 0.96234803 | 0.940486033 |
| First Quartile (Q1) | 0.999972893 | 0.946449628 | 0.964130332 | 0.950834226 |
| Median Value | 0.99997967 | 0.996204985 | 0.965878749 | 0.959162928 |
| Third Quartile(Q3) | 0.999986446 | 0.998759843 | 0.967457747 | 0.966719074 |
| Max Value | 0.999986446 | 0.999119014 | 0.969206164 | 0.976863962 |
| Box 1-hidden (Q1) | 0.999972893 | 0.946449628 | 0.964130332 | 0.950834226 |
| Box 2 (Median - Q1) | 6.77681E-06 | 0.049755357 | 0.001748418 | 0.008328703 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Box 3 (Q3-Median) | 6.77681E-06 | 0.002554858 | 0.001578997 | 0.007556146 |
| Whisker Top (Max- Q3) | 0 | 0.000359171 | 0.001748418 | 0.010144888 |
| Whisker Bottom (Q1- Min) | 0 | 0.129911495 | 0.001782302 | 0.010348193 |

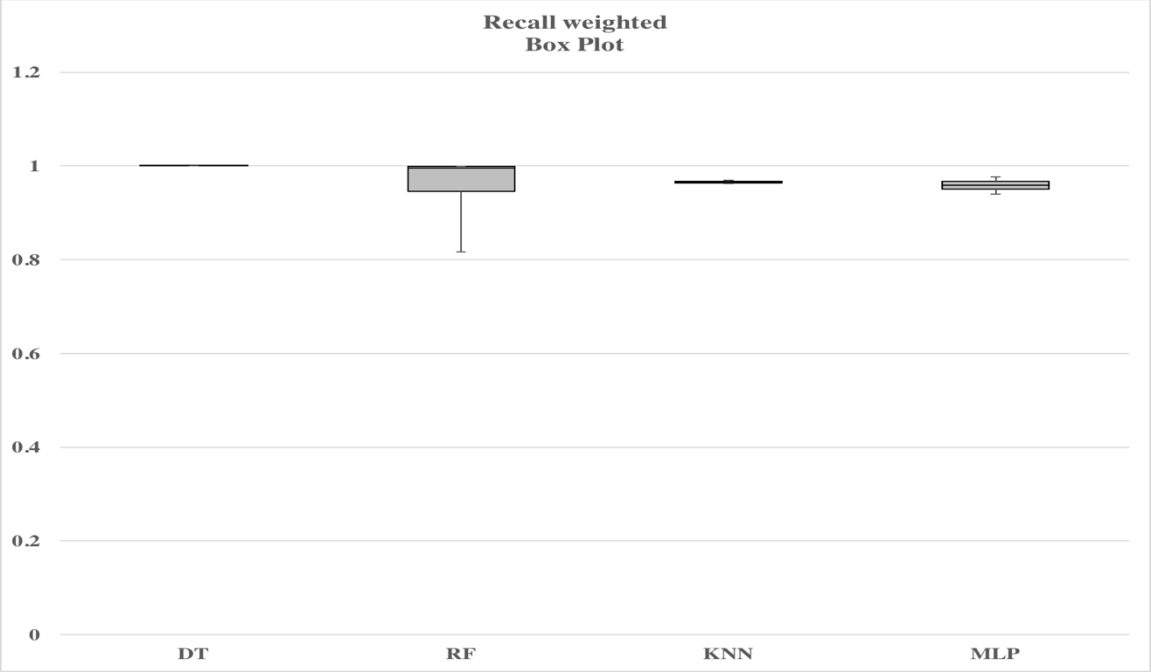


Figure 4.127. Recall Weighted Box Plot for BMI Dataset Using All Features

#### 4.7.1.3 Best Model

The following diagram shows the best model created varying only one coefficient of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithms using all features for BMI dataset.

| Algorithm | Accuracy | | Precision Micro | | Precision Macro | | Precision Weighted | | Recall Micro | | Recall Macro | | Recall Weighted | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value |
| Decission Tree | max_depth: 1 | 0.999986446 | max_depth: 1 | 0.999986446 | max_depth: 1 | 0.999963073 | max_depth: 1 | 0.999986451 | max_depth: 1 | 0.999986446 | max_depth: 1 | 0.999991701 | max_depth: 1 | 0.999986446 |
| Random Forest | max_depth: 14 | 0.99119014 | max_depth: 14 | 0.99119014 | max_depth: 16 | 0.999206103 | max_depth: 14 | 0.99119204 | max_depth: 14 | 0.99119014 | max_depth: 14 | 0.997856778 | max_depth: 14 | 0.99119014 |
| KNN | n_neighbors: 5 | 0.969206164 | n_neighbors: 5 | 0.969206164 | n_neighbors: 4 | 0.963638339 | n_neighbors: 5 | 0.968880655 | n_neighbors: 5 | 0.969206164 | n_neighbors: 1 | 0.941731289 | n_neighbors: 5 | 0.969206164 |
| MLP | max_iter: 40000 | 0.97565769 | max_iter: 70000 | 0.973001179 | max_iter: 80000 | 0.975958119 | max_iter: 100000 | 0.978893515 | max_iter: 140000 | 0.972621678 | max_iter: 60000 | 0.958475168 | max_iter: 70000 | 0.976863962 |

Figure 4.128. Best Model for BMI Dataset Using All Features

### 4.7.2 Using Transfer Learning

In transfer learning technique, top 10 important features were identified during transfer learning using decision tree. And these top 10 features were only used for training all the models of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithm to demonstrate the transfer learning in this experiment. The following sections shows the comparison between the models.

#### 4.5.2.1 Line Graph

Following line diagram shows the comparison of different models of each algorithm based on evaluation metrics.

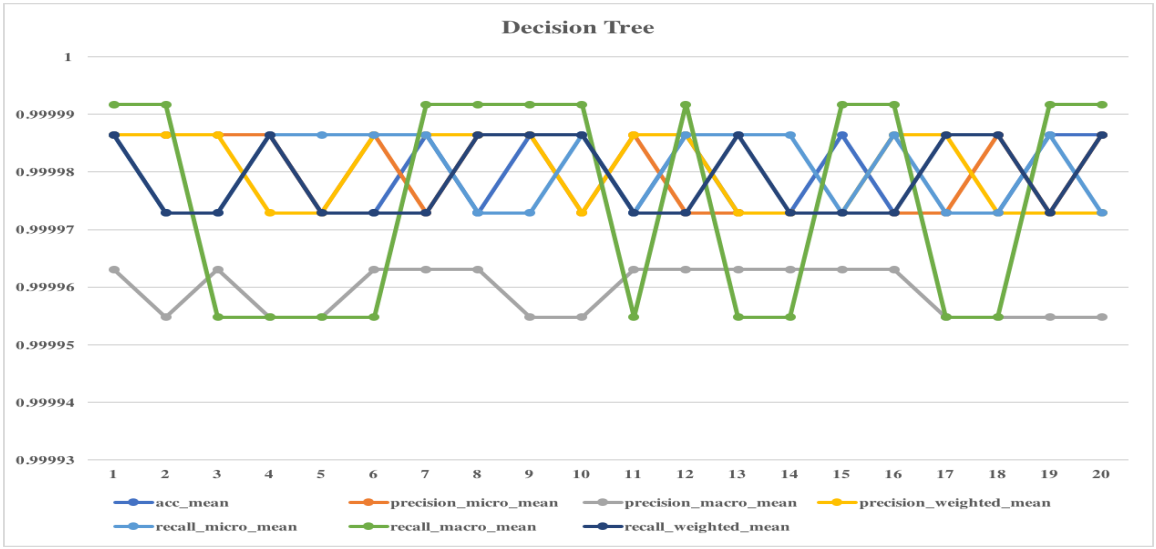


Figure 4.129. Line Graph of Decision Tree with Varying Max_Depth for BMI Dataset Using Transfer Learning

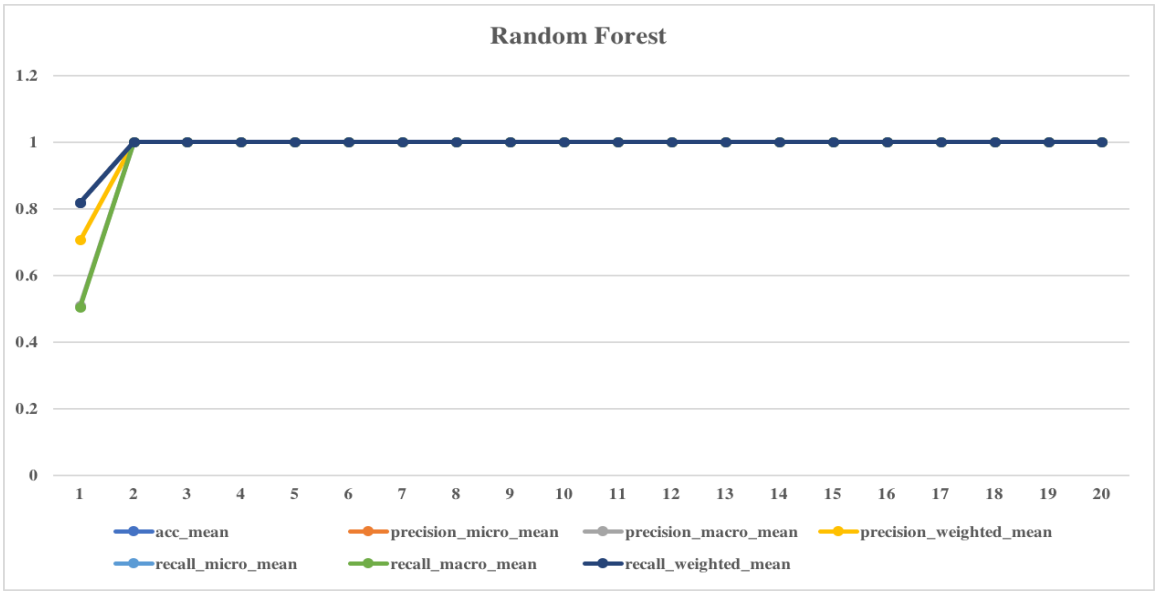


Figure 4.130. Line Graph of Random Forest with Varying Max_Depth for BMI Dataset Using Transfer Learning

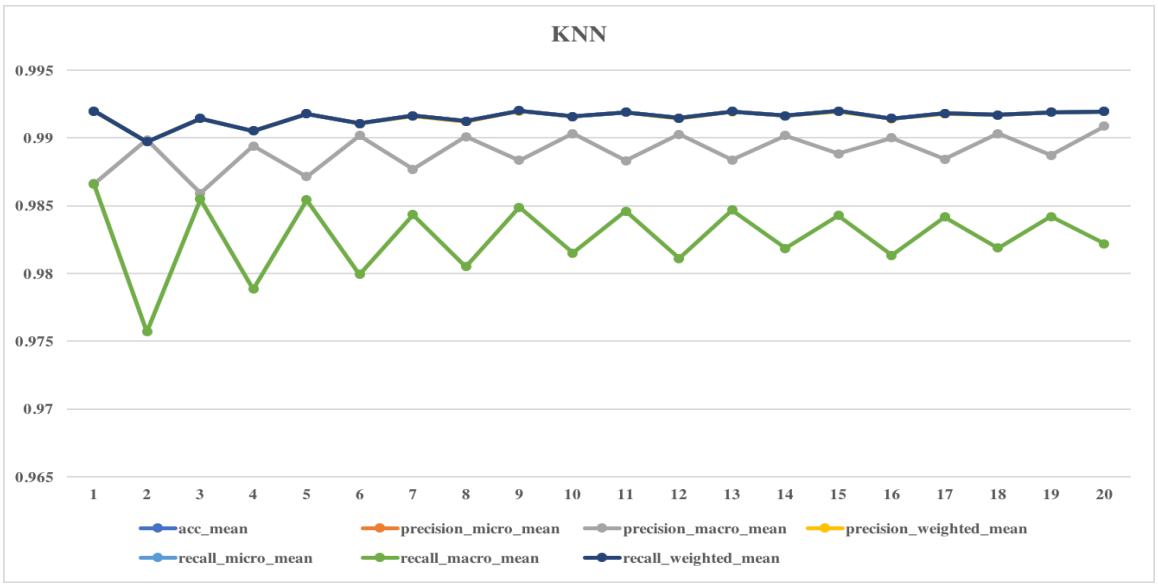


Figure 4.131. Line Graph of KNN with Varying N_Neighbor for BMI Dataset Using Transfer Learning

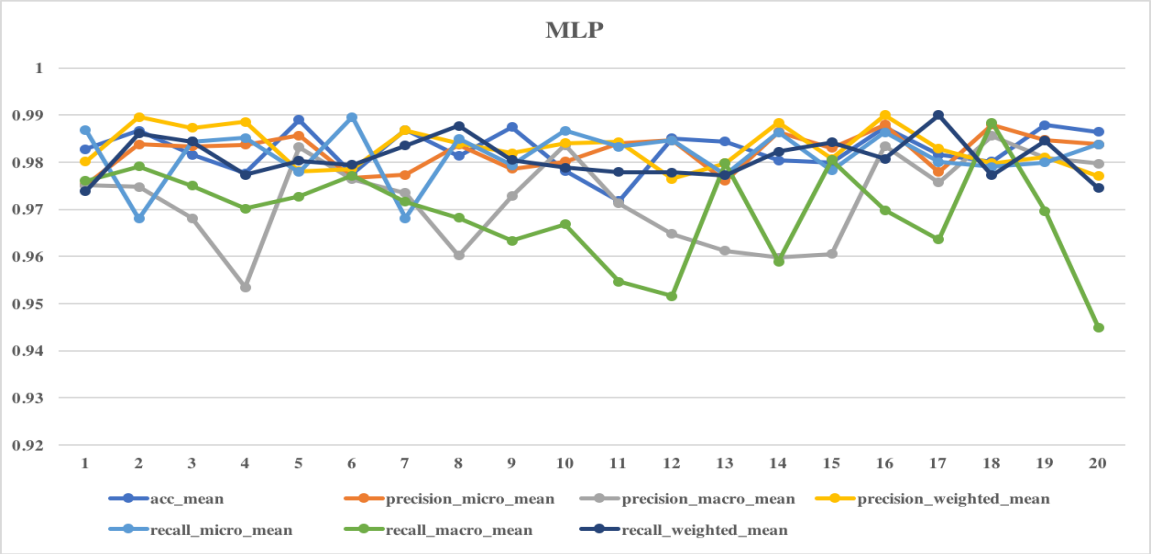


Figure 4.132. Line Graph of MLP with Varying Max_Iteration for BMI Dataset Using Transfer Learning

#### 4.7.2.2 Box Plot

Following box diagram, shows the comparison of each algorithm based on grid search with 5-fold cross validation.

Table 4.66. Accuracy Value for BMI Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.999972893 | 0.817812174 | 0.989726352 | 0.971767799 |
| First Quartile (Q1) | 0.999972893 | 0.999986446 | 0.991444274 | 0.980082948 |
| Median Value | 0.999986446 | 0.999986446 | 0.991678074 | 0.982170206 |
| Third Quartile(Q3) | 0.999986446 | 0.999986446 | 0.991918651 | 0.986683564 |
| Max Value | 0.999986446 | 0.999986446 | 0.992003361 | 0.988994457 |
| Box 1-hidden (Q1) | 0.999972893 | 0.999986446 | 0.991444274 | 0.980082948 |
| Box 2 (Median - Q1) | 1.35536E-05 | 0 | 0.0002338 | 0.002087258 |
| Box 3 (Q3- Median) | 0 | 0 | 0.000240577 | 0.004513357 |
| Whisker Top (Max- Q3) | 0 | 0 | 8.47102E-05 | 0.002310893 |

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Whisker Bottom (Q1- Min) | 0 | 0.182174273 | 0.001717922 | 0.008315149 |

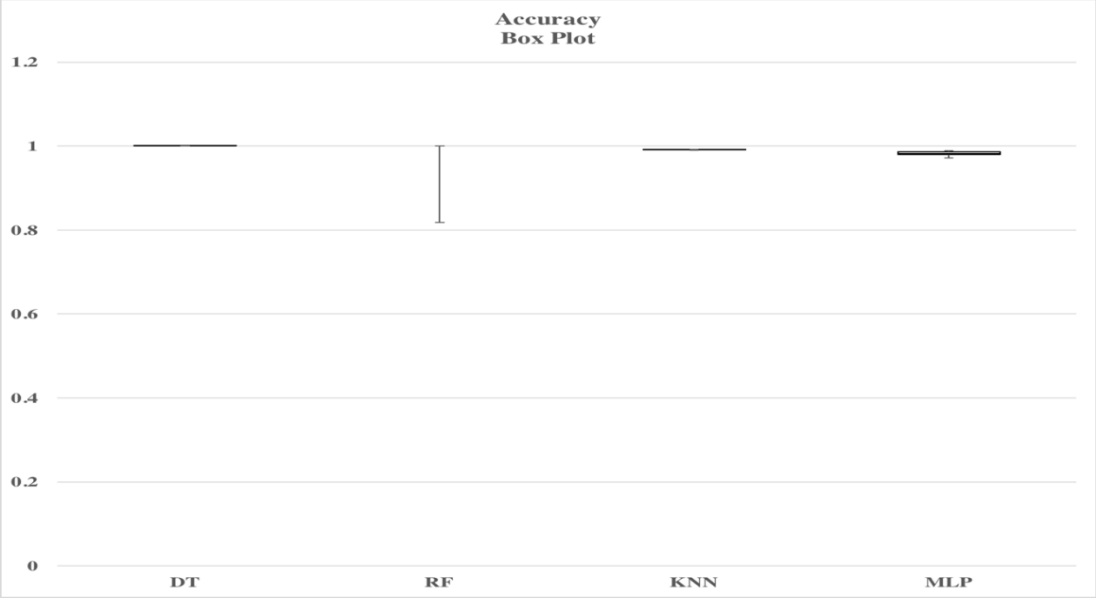


Figure 4.133. Accuracy Box Plot for BMI Dataset Using Transfer Learning

Table 4.67. Precision Macro Value for BMI Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.999954774 | 0.508791204 | 0.985927001 | 0.953481475 |
| First Quartile (Q1) | 0.999954774 | 0.999963073 | 0.988354806 | 0.963874182 |
| Median Value | 0.999963073 | 0.999963073 | 0.989138384 | 0.974112251 |
| Third Quartile(Q3) | 0.999963073 | 0.999963073 | 0.990173181 | 0.980022375 |
| Max Value | 0.999963073 | 0.999963073 | 0.990881232 | 0.985598784 |
| Box 1-hidden (Q1) | 0.999954774 | 0.999963073 | 0.988354806 | 0.963874182 |
| Box 2 (Median - Q1) | 8.29921E-06 | 0 | 0.000783577 | 0.010238069 |
| Box 3 (Q3- Median) | 0 | 0 | 0.001034797 | 0.005910124 |
| Whisker Top (Max- Q3) | 0 | 0 | 0.000708051 | 0.005576409 |
| Whisker Bottom (Q1- Min) | 0 | 0.491171868 | 0.002427806 | 0.010392708 |

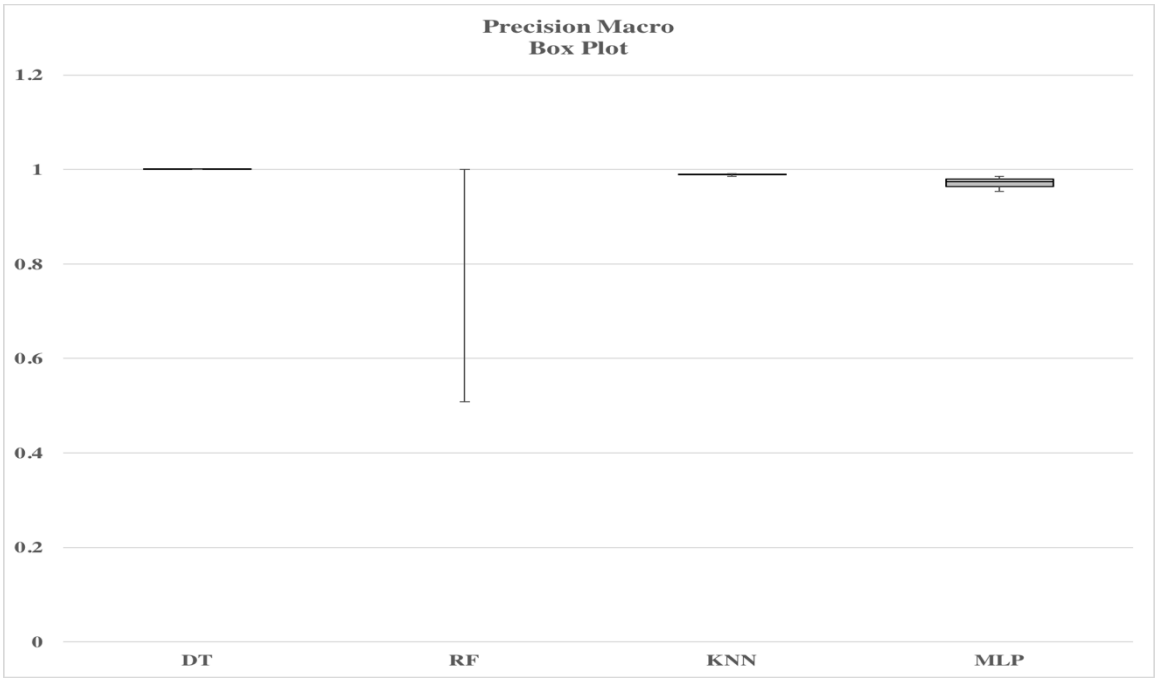


Figure 4.134. Precision Macro Box Plot for BMI Dataset Using Transfer Learning

Table 4.68. Precision Micro Value for BMI Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.999972893 | 0.817812174 | 0.989726352 | 0.975142652 |
| First Quartile (Q1) | 0.999972893 | 0.999986446 | 0.991444274 | 0.978415852 |
| Median Value | 0.99997967 | 0.999986446 | 0.991678074 | 0.983661105 |
| Third Quartile(Q3) | 0.999986446 | 0.999986446 | 0.991918651 | 0.984640355 |
| Max Value | 0.999986446 | 0.999986446 | 0.992003361 | 0.987950827 |
| Box 1-hidden (Q1) | 0.999972893 | 0.999986446 | 0.991444274 | 0.978415852 |
| Box 2 (Median - Q1) | 6.77681E-06 | 0 | 0.0002338 | 0.005245253 |
| Box 3 (Q3- Median) | 6.77681E-06 | 0 | 0.000240577 | 0.000979249 |
| Whisker Top (Max- Q3) | 0 | 0 | 8.47102E-05 | 0.003310473 |
| Whisker Bottom (Q1- Min) | 0 | 0.182174273 | 0.001717922 | 0.0032732 |

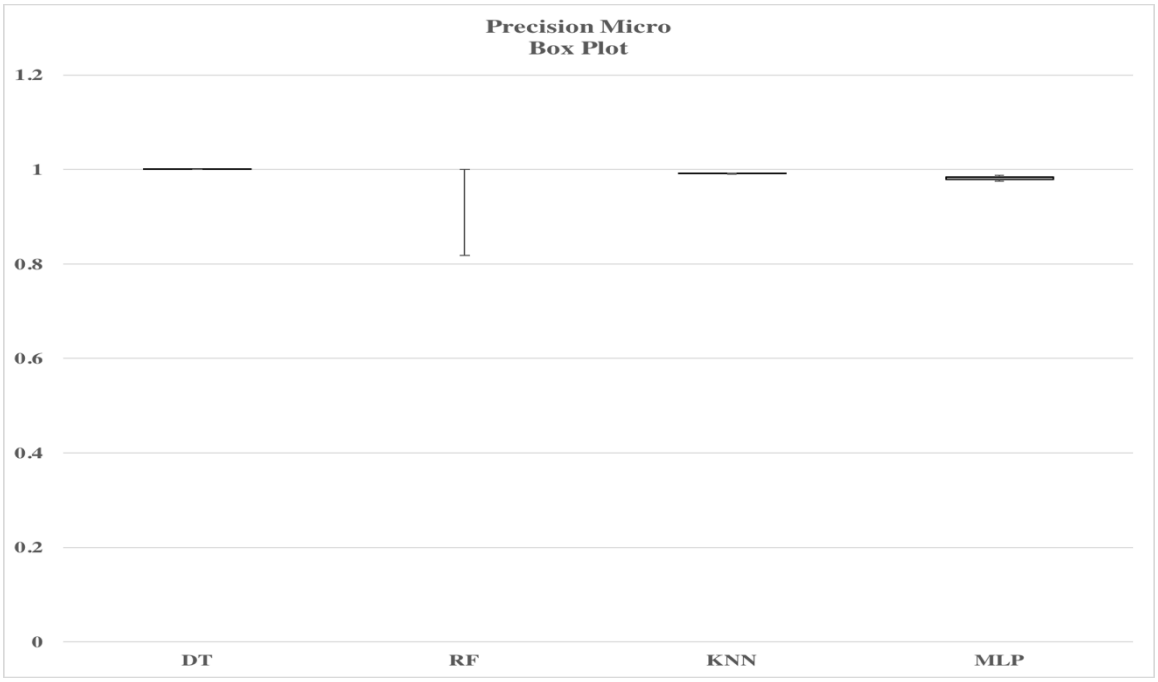


Figure 4.135. Precision Micro Box Plot for BMI Dataset Using Transfer Learning

Table 4.69. Precision Weighted Value for BMI Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.999972899 | 0.704279102 | 0.989729743 | 0.976440803 |
| First Quartile (Q1) | 0.999972899 | 0.999986451 | 0.991432003 | 0.979795493 |
| Median Value | 0.999986451 | 0.999986451 | 0.99166216 | 0.98236196 |
| Third Quartile(Q3) | 0.999986451 | 0.999986451 | 0.991901476 | 0.986901883 |
| Max Value | 0.999986451 | 0.999986451 | 0.991986998 | 0.990028778 |
| Box 1-hidden (Q1) | 0.999972899 | 0.999986451 | 0.991432003 | 0.979795493 |
| Box 2 (Median - Q1) | 1.35525E-05 | 0 | 0.000230157 | 0.002566467 |
| Box 3 (Q3- Median) | 0 | 0 | 0.000239317 | 0.004539923 |
| Whisker Top (Max- Q3) | 0 | 0 | 8.55217E-05 | 0.003126894 |
| Whisker Bottom (Q1- Min) | 0 | 0.295707349 | 0.001702261 | 0.003354691 |

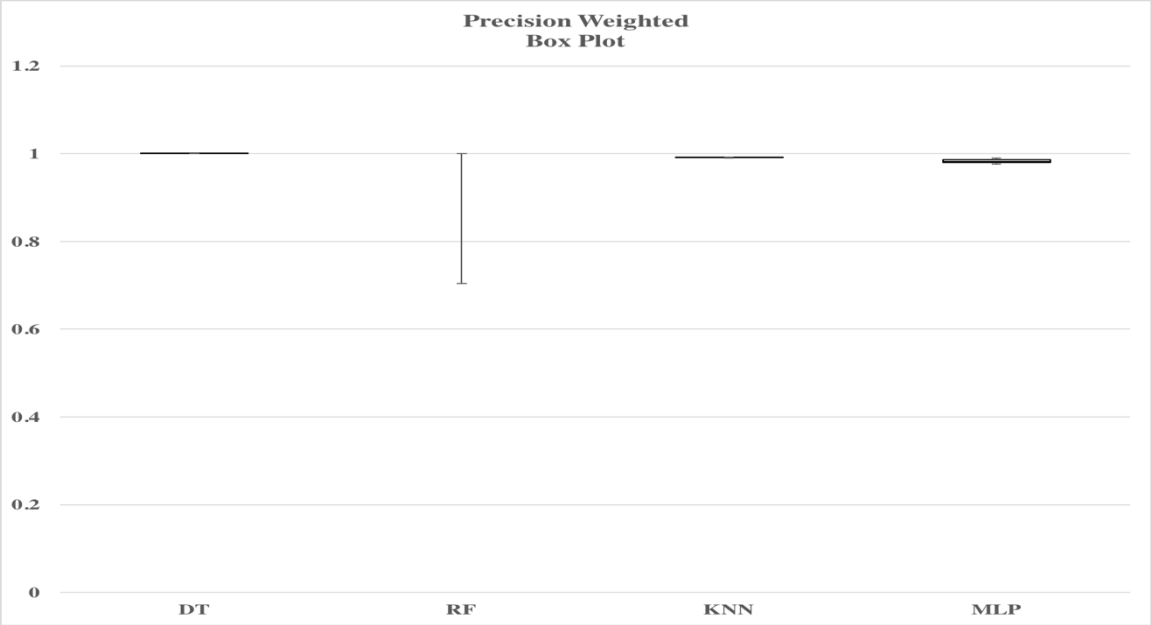


Figure 4.136. Precision Weighted Box Plot for BMI Dataset Using Transfer Learning

Table 4.70. Recall Macro Value for BMI Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.999954771 | 0.503472432 | 0.975694967 | 0.94494752 |
| First Quartile (Q1) | 0.999954771 | 0.999991701 | 0.981276093 | 0.9635125 |
| Median Value | 0.999991701 | 0.999991701 | 0.983176102 | 0.969978638 |
| Third Quartile(Q3) | 0.999991701 | 0.999991701 | 0.984612366 | 0.976311201 |
| Max Value | 0.999991701 | 0.999991701 | 0.986580829 | 0.988393853 |
| Box 1-hidden (Q1) | 0.999954771 | 0.999991701 | 0.981276093 | 0.9635125 |
| Box 2 (Median - Q1) | 3.69296E-05 | 0 | 0.001900009 | 0.006466138 |
| Box 3 (Q3- Median) | 0 | 0 | 0.001436265 | 0.006332563 |
| Whisker Top (Max- Q3) | 0 | 0 | 0.001968462 | 0.012082652 |
| Whisker Bottom (Q1- Min) | 0 | 0.496519269 | 0.005581126 | 0.01856498 |

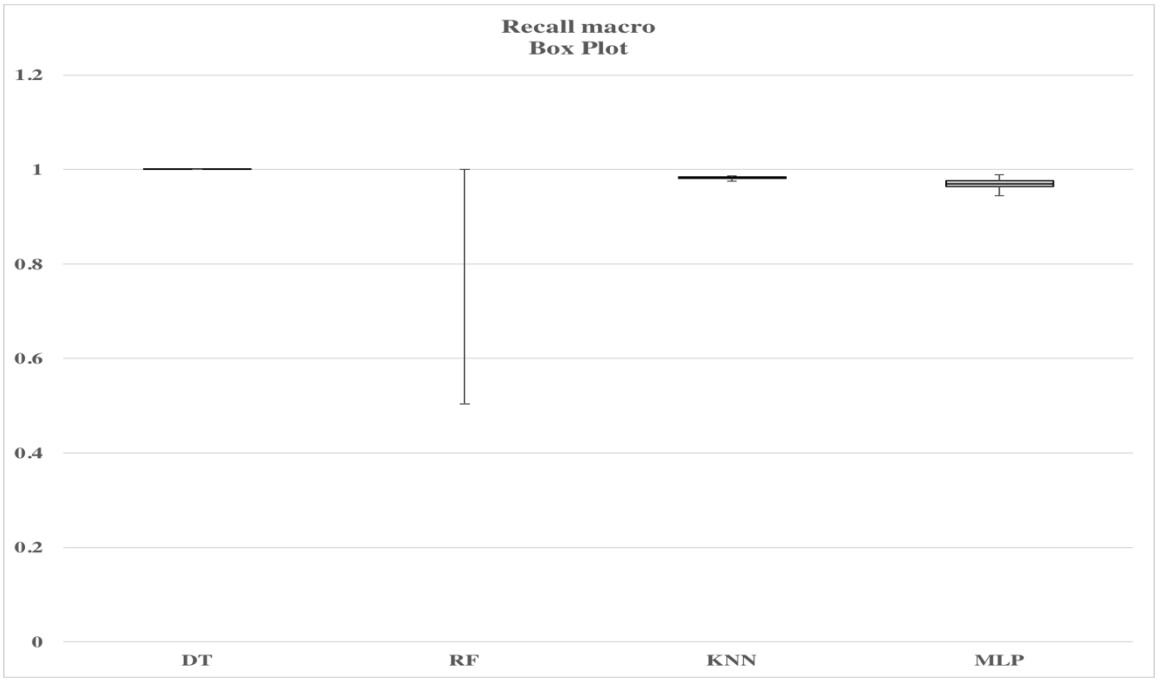


Figure 4.137. Recall Macro Box Plot for BMI Dataset Using Transfer Learning

Table 4.71. Recall Macro Value for BMI Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.999972893 | 0.817812174 | 0.989726352 | 0.967986338 |
| First Quartile (Q1) | 0.999972893 | 0.999986446 | 0.991444274 | 0.978842792 |
| Median Value | 0.999986446 | 0.999986446 | 0.991678074 | 0.983518792 |
| Third Quartile(Q3) | 0.999986446 | 0.999986446 | 0.991918651 | 0.985423076 |
| Max Value | 0.999986446 | 0.999986446 | 0.992003361 | 0.989536602 |
| Box 1-hidden (Q1) | 0.999972893 | 0.999986446 | 0.991444274 | 0.978842792 |
| Box 2 (Median - Q1) | 1.35536E-05 | 0 | 0.0002338 | 0.004676001 |
| Box 3 (Q3- Median) | 0 | 0 | 0.000240577 | 0.001904284 |
| Whisker Top (Max- Q3) | 0 | 0 | 8.47102E-05 | 0.004113525 |
| Whisker Bottom (Q1- Min) | 0 | 0.182174273 | 0.001717922 | 0.010856454 |

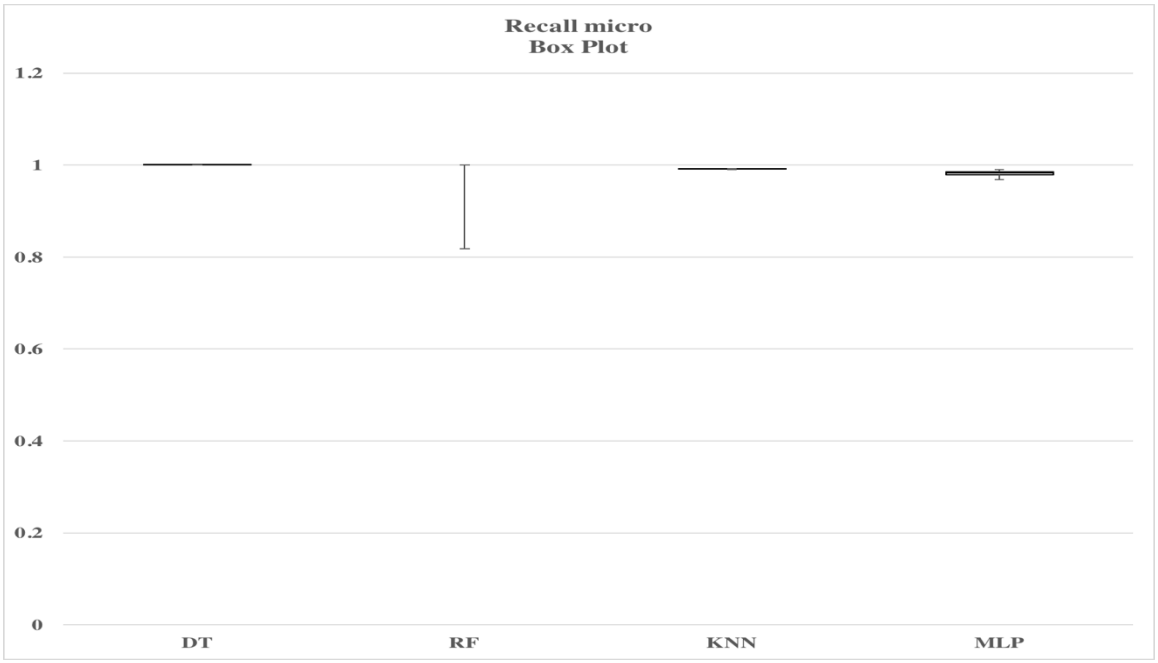


Figure 4.138. Recall Micro Box Plot for BMI Dataset Using Transfer Learning

Table 4.72. Recall Weighted Value for BMI Dataset Using Transfer Learning

| Parameter | Decision Tree | Random Forest | KNN | MLP |
|---|---|---|---|---|
| Min Value | 0.999972893 | 0.817812174 | 0.989726352 | 0.973800843 |
| First Quartile (Q1) | 0.999972893 | 0.999986446 | 0.991444274 | 0.977660238 |
| Median Value | 0.999972893 | 0.999986446 | 0.991678074 | 0.980381128 |
| Third Quartile(Q3) | 0.999986446 | 0.999986446 | 0.991918651 | 0.984240523 |
| Max Value | 0.999986446 | 0.999986446 | 0.992003361 | 0.989983871 |
| Box 1-hidden (Q1) | 0.999972893 | 0.999986446 | 0.991444274 | 0.977660238 |
| Box 2 (Median - Q1) | 0 | 0 | 0.0002338 | 0.00272089 |
| Box 3 (Q3- Median) | 1.35536E-05 | 0 | 0.000240577 | 0.003859395 |
| Whisker Top (Max- Q3) | 0 | 0 | 8.47102E-05 | 0.005743349 |
| Whisker Bottom (Q1- Min) | 0 | 0.182174273 | 0.001717922 | 0.003859395 |

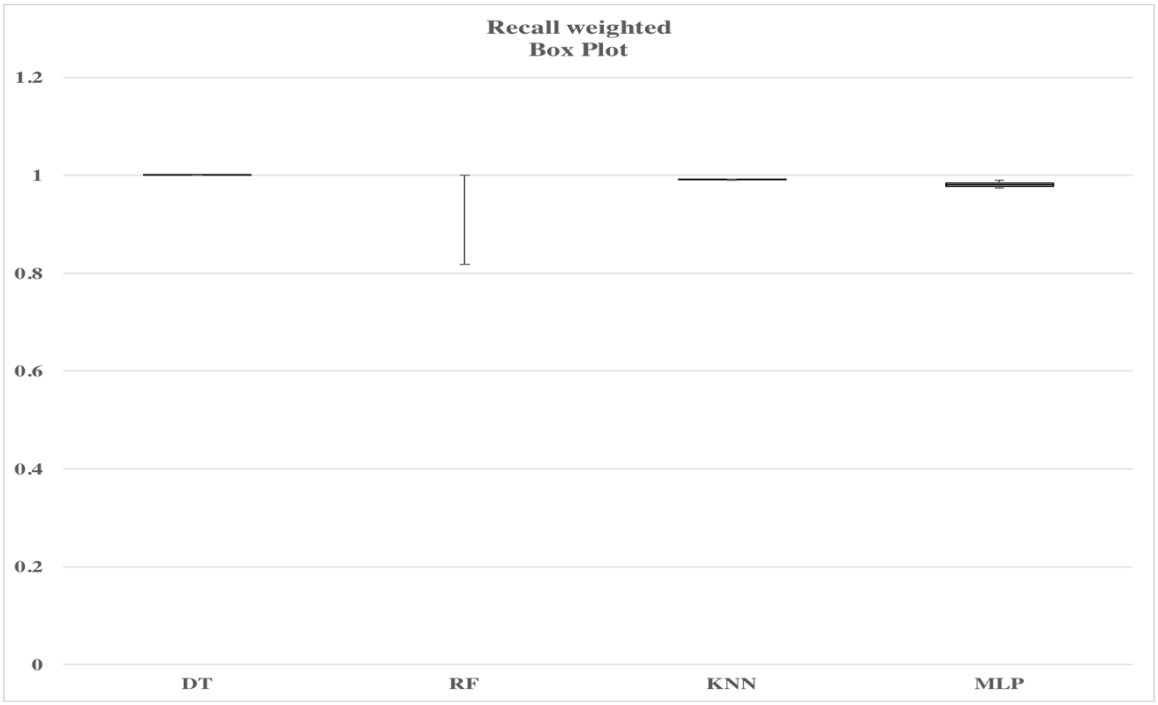


Figure 4.139. Recall Weighted Box Plot for BMI Dataset Using Transfer Learning

#### 4.7.2.3 Best Model

The following diagram shows the best model created varying only one coefficient of Decision Tree, Random Forest, K-Nearest Neighbor and MLP algorithms using transfer learning for BMI dataset.

| Algorithm | Accuracy | | Precision Micro | | Precision Macro | | Precision Weighted | | Recall Micro | | Recall Macro | | Recall Weighted | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value |
| Decission Tree | max_depth: 1 | 0.999986446 | max_depth: 1 | 0.999986446 | max_depth: 1 | 0.999963073 | max_depth: 1 | 0.999986451 | max_depth: 1 | 0.999986446 | max_depth: 1 | 0.999991701 | max_depth: 1 | 0.999986446 |
| Random Forest | max_depth: 2 | 0.999986446 | max_depth: 2 | 0.999986446 | max_depth: 2 | 0.999963073 | max_depth: 2 | 0.999986451 | max_depth: 2 | 0.999986446 | max_depth: 2 | 0.999991701 | max_depth: 2 | 0.999986446 |
| KNN | n_neighbors: 9 | 0.992003361 | n_neighbors: 9 | 0.992003361 | n_neighbors: 20 | 0.990881232 | n_neighbors: 9 | 0.991986998 | n_neighbors: 9 | 0.992003361 | n_neighbors: 1 | 0.986580829 | n_neighbors: 9 | 0.992003361 |
| MLP | max_iter: 50000 | 0.988994457 | max_iter: 180000 | 0.987950827 | max_iter: 18000 | 0.985598784 | max_iter: 160000 | 0.990028778 | max_iter: 60000 | 0.989536602 | max_iter: 180000 | 0.988393853 | max_iter: 170000 | 0.989983871 |

Figure 4.140. Best Model for BMI Dataset Using Transfer Learning

### 4.7.3 Methodology and Algorithm Comparison Based on Accuracy for BMI Dataset

In this section, we compare all the 2-methodologies used with BMI dataset, following table shows the comparisons between best models for each methodology and each machine learning algorithms.

Table 4.73. Accuracy Based Comparisons of Best Model for BMI Dataset

| **Features Used for Training** | **Best Model Accuracy with Grid Search Evaluation** | | | |
|---|---|---|---|---|
| | **Decision Tree** | **Random Forest** | **KNN** | **MLP** |
| **All 118 features** | **0.999986446** | **0.999119014** | **0.969206164** | **0.97565769** |
| **Transfer learning** | **0.999986446** | **0.999986446** | **0.992003361** | **0.988994457** |

The table 71 shows that the transfer learning methodology best model has better or almost same accuracy then all features methodology.

# CHAPTER V

# CONCLUSION AND FUTURE WORK

## 5.1 Conclusion

We compared the performance of the Decision Tree, Random Forest, K-Nearest Neighbor and Multilayer Perceptron machine learning algorithms for two different datasets, one with heart disease dataset and another with readmission dataset. These comparisons were based on the technique and number of features used to train the models. In traditional approach where all available features of the dataset were used to train the model, the best model of the Decision Tree outperformed with an accuracy of 99.84% for heart disease dataset, and followed by Random forest, KNN and MLP whereas for readmission dataset and BMI dataset has accuracy of 57.38% and 99.9986%.

In the transfer learning technique, the top ten important features were identified out of all the other features using Decision Tree, and these important features were used to train the models. This technique showed that all algorithms performance was almost the same or the best in some cases then the traditional approach [section evaluation diagram]. Here, also the best model of the decision tree outperformed with accuracy of 99.91% for heart disease dataset whereas 57.38% and 99.986% for readmission dataset and BMI dataset.

We also did experiments with the training models with expert suggested features for heart disease dataset and readmission dataset, which showed the performance of the model dropped then transfer learning. It had highest accuracy of 85.9% and 53.98%by Random forest algorithm model for heart disease dataset and for readmission dataset.

We also trained the models, combining the expert suggested features and the top 10 important features we identified during transfer learning for heart disease dataset and readmission dataset. This experiment advocated that the expert suggested missed the features correlated with output as the performance of the models increased more than in previous experiment, where we used only the suggested features. The best model of the Decision Tree algorithm had the accuracy of 99.91% for heart disease dataset whereas the best model of Random forest had the accuracy of 57.26% for readmission dataset.

With all these experiments, we concluded that all the features in dataset might not be correlated with the outputs. So, if we know the minimum number of important features that are correlated with the output, then it helps in creating the models which have almost the same or better performance than when we used all the features. From our experiments, we also concluded that the important features identified during transfer learning help in reducing time and complexity in training the model, by using only correlated features of output.

For future work, we would like to find the best model out of different models created by modifying all the coefficients of algorithm other than just one. We would also like to find the minimal number for important features needed to have the same or better performance than using all the features. We think transfer learning technique helps in reducing the time and complexity for training the models, and it achieves the same or better performance than traditional machine learning techniques.

REFERENCES


1. The top 10 cause of death in 2015, World health Organization, (http://www.who.int/mediacentre/factsheets/fs310/en/)

2. Heart Disease and Stroke statistics-2017 update, American Heart Association. http://professional.heart.org/professional/ScienceNews/UCM_491264_Heart-Disease-and-Stroke-Statistics---2017-Update.jsp

3. Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10):1345–1359, 2010.

4. Witten IH, Frank E. Data mining, practical machine learning tools and techniques. 3rd ed. San Francisco: Morgan Kaufmann Publishers; 2011.

5. Avrim L.Blum and Pat Langley. Selection of relevant features and exmples in machine learning. Artificial Intelligence, 97(1-2):245-271, 1997

6. John, G.H., Kohavi, R., and Pfleger, K. (1994). Irrelevant Features and the Subset Selection Problem. In: Cohen, W.W., & Hirsh, H. (eds). ICML 1994, pp. 121-129.

7. Erico Guizzo. How Google's self-driving car works. IEEE Spectrum Online, October, 18, 2011.

8. Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. Machine Learning, 46(1-3):389–422, 2002.

9. Jyoti Soni, Ujma Ansari, Dipesh Sharma and Sunita Soni. Article: Predictive Data Mining for Medical Diagnosis: An Overview of Heart Disease Prediction. International Journal of Computer Applications 17(8):43-48, March 2011

10. Pat Langley and Wayne Iba. Average-case analysis of a nearest neighbor algorithm, April 13,1993.

11. Margaret A Shipp, Ken N Ross, Pablo Tamayo, Andrew P Weng, Jeffery L Kutok, et al. Diffuse large b-cell lymphoma outcome prediction by geneexpression profiling and supervised machine learning. Nature Medicine, 8(1):68–74, 2002.

12. Robert R Trippi and Efraim Turban. Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real World Performance. McGraw-Hill, Inc., 1992.

13. A. Argyriou, T. Evgeniou, and M. Pontil, “Multi-task feature learning,” in Proceedings of the 19th Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 2007, pp. 41–48.

14. X Liao, Y Xue, and L Carin. Logistic regression with an auxiliary data source. In Proceedings of the 22nd International Conference on Machine Learning, pages 505–512. ACM, 2005.

15. P Luo, F Zhuang, H Xiong, Y Xiong, and Q He. Transfer learning from multiple source domains via consensus regularization. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, pages 103–112. ACM, 2008.

16. Wolf Kienzle and Kumar Chellapilla. Personalized handwriting recognition via biased regularization. In Proceedings of the 24th International Conference on Machine Learning (ICML), pages 457- 464. ACM, 2006.

17. Zhongqi Lu, Weike Pan, Evan Wei Xiang, Qiang Yang, Lili Zhao, and ErHeng Zhong. Selective transfer learning for cross domain recommendation. In Proceedings of the 2013 SIAM International Conference on Data Mining, pages 641-649. SDM, 2013.

18. David Pardoe and Peter Stone. Boosting for regression transfer. In Proceedings of the 27th International Conference on Machine learning (ICML), pages 863–870. ACM, 2010.

19. Jonathan Baxter. A model of inductive bias learning. J. Artif. Intell. Res.(JAIR), 12:149–198, 2000.

20. Ann L Brown and Mary Jo Kane. Preschool children can learn to transfer: Learning to learn and learning from example. Cognitive Psychology, 20(4):493–523, 1988.

21. L Duan, I W Tsang, D Xu, and T S Chua. Domain adaptation from multiple sources via auxiliary classifiers. In Proceedings of the 26th Annual International Conference on Machine Learning, 2009.

22. Toshihiro Kamishima, Masahiro Hamasaki, and Shotaro Akaho. Trbagg: A simple transfer learning method and its application to personalization in collaborative tagging. In Proceedings of the 9th International Conference on Data Mining (ICDM), pages 219-228. IEEE, 2009.

23. R K Ando and T Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. The Journal of Machine Learning Research, 6:1817–1853, 2005.

24. A Argyriou, T Evgeniou, and M Pontil. Convex multi-task feature learning. Machine Learning, 73(3):243–272, 2008.

25. Mahsa Baktashmotlagh, Mehrtash T Harandi, Brian C Lovell, and Mathieu Salzmann. Unsupervised domain adaptation by domain invariant projection. In International Conference on Computer Vision (ICCV), pages 769–776. IEEE, 2013.

26. John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pages 120–128.

27. Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In Proceedings of the 11th Annual Conference on Computational Learning Theory, pages 92–100. ACM, 1998.

28. Raymond Board and Leonard Pitt. Semi-supervised learning. Machine Learning, 4(1):41–65, 1989.

29. E V Bonilla, K M Chai, and C KI Williams. Multi-task Gaussian process prediction. In NIPS, volume 20, 2007.

30. Olivier Chapelle, Bernhard Sch¨olkopf, Alexander Zien, et al. Semi-supervised Learning, volume 2. MIT Press, Cambridge, 2006.

31. W Dai, Q Yang, G R Xue, and Y Yu. Boosting for transfer learning. In Proceedings of the 24th International Conference on Machine Learning, 2007.

32. H Daum´e III. Frustratingly easy domain adaptation. In Conference of the Association for Computational Linguistics (ACL), 2007.

33. Hal Daum´e III and Daniel Marcu. Domain adaptation for statistical classi- fiers. J. Artif. Intell. Res.(JAIR), 26:101–126, 2006.

34. M Dredze, A Kulesza, and K Crammer. Multi-domain learning by confidenceweighted parameter combination. Machine Learning, 79(1-2):123–149, 2010.

35. L Duan, I W Tsang, D Xu, and T S Chua. Domain adaptation from multiple sources via auxiliary classifiers. In Proceedings of the 26th Annual International Conference on Machine Learning, 2009. 52 Technion - Computer Science Department - M.Sc. Thesis MSC-2016-02 - 2016

36. Lixin Duan, Ivor W Tsang, and Dong Xu. Domain transfer multiple kernel learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(3):465–479, 2012.

37. Lixin Duan, Ivor W Tsang, Dong Xu, and Stephen J Maybank. Domain transfer svm for video concept detection. In Computer Vision and Pattern Recognition (CVPR), pages 1375–1381. IEEE, 2009.

38. Lixin Duan, Dong Xu, and Shih-Fu Chang. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In Computer Vision and Pattern Recognition (CVPR), pages 1338–1345. IEEE, 2012.

39. Lixin Duan, Dong Xu, and Ivor W Tsang. Domain adaptation from multiple sources: A domain-dependent regularization approach. IEEE Transactions on Neural Networks and Learning Systems, 23(3):504–518, 2012.

40. Lixin Duan, Dong Xu, IW-H Tsang, and Jiebo Luo. Visual event recognition in videos by learning from web data. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(9):1667–1680, 2012.

41. Eric Eaton, Marie desJardins, and Terran Lane. Modeling transfer relationships between learning tasks for improved inductive transfer. Machine Learning and Knowledge Discovery in Databases, pages 317–332, 2008.

42. Victor Eruhimov, Vladimir Martyanov, and Aleksey Polovinkin. Transferring knowledge by prior feature sampling. In FSDM, pages 135–147, 2008.

43. A Evgeniou and Massimiliano Pontil. Multi-task feature learning. Advances in Neural Information Processing Systems, 19:41, 2007.

44. T Evgeniou and M Pontil. Regularized multi–task learning. In Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004.

45. Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In International Conference on Computer Vision (ICCV), pages 2960–2967. IEEE, 2013.

46. Norberto A Goussies, Sebasti´an Ubalde, and Marta Mejail. Transfer learning decision forests for gesture recognition. Journal of Machine Learning Research, 15:3667–3690, 2014.

47. Maayan Harel and Shie Mannor. Learning from multiple outlooks. In Proceedings of the 28th International Conference on Machine Learning (ICML), pages 401–408. ACM, 2011.

48. Jing Jiang. A literature survey on domain adaptation of statistical classifiers. URL: http://sifaka.cs.uiuc.edu/jiang4/domainadaptation/survey, 2008.

49. Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in NLP. In ACL, volume 7, pages 264–271, 2007.

50. Wei Jiang, Eric Zavesky, Shih-Fu Chang, and Alex Loui. Cross-domain learning methods for high-level visual concept classification. In Proceedings of the 15th IEEE International Conference on Image Processing (ICIP), 2008.

51. Luo Jie, Tatiana Tommasi, and Barbara Caputo. Multiclass transfer learning from unconstrained priors. In International Conference on Computer Vision (ICCV), pages 1863–1870. IEEE, 2011.

52. Neil D Lawrence and John C Platt. Learning to learn with the informative vector machine. In Proceedings of the 21st International Conference on Machine Learning (ICML), page 65. ACM, 2004.

53. Omer Levy and Shaul Markovitch. Teaching machines to learn by metaphors. In Proceedings of the 26th Conference on Artificial Intelligence, pages 991– 997. AAAI, 2012.

54. K Lounici, M Pontil, AB Tsybakov, and SAVD Geer. Taking advantage of sparsity in multi-task learning. In Proceedings of the 22nd Annual Conference on Learning Theory (COLT). ACL, 2009.

55. Sinno Jialin Pan, James T Kwok, and Qiang Yang. Transfer learning via dimensionality reduction. In Proceedings of the 23rd Conference on Artificial Intelligence, pages 677–682. AAAI, 2008.

56. Lorien Y Pratt, Jack Mostow, Candace A Kamm, and Ace A Kamm. Direct transfer of learned information among neural networks. In Proceedings of the 9th National Conference on Artificial Intelligence, pages 584–589. AAAI, 1991.

57. Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In Proceedings of the 24th International Conference on Machine Learning (ICML), pages 759–766. ACM, 2007.

58. Achim Rettinger, Martin Zinkevich, and Michael Bowling. Boosting expert ensembles for rapid concept recall. In Proceedings of the 21st National Conference on Artificial Intelligence, volume 21, page 464. AAAI, 2006.

59. Erik Rodner and Joachim Denzler. Learning with few examples using a constrained gaussian prior on randomized trees. In Proceedings of the Vision, Modeling, and Visualization Conference (VMV), pages 159–168. Pro Universitate, 2008.

60. Erik Rodner and Joachim Denzler. Learning with few examples by transferring feature relevance. In Pattern Recognition, pages 252–261. Springer, 2009.

61. Erik Rodner and Joachim Denzler. Learning with few examples for binary and multiclass classification using regularization of randomized trees. Pattern Recognition Letters, 32(2):244–251, 2011.

62. Ulrich R¨uckert and Stefan Kramer. Kernel-based inductive transfer. In Machine Learning and Knowledge Discovery in Databases, pages 220–233. Springer, 2008.

63. Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In Proceedings of the European Conference on Computer Vision (ECCV), pages 213–226. Springer, 2010.

64. A. Schwaighofer, V. Tresp, and K. Yu. Hierarchical bayesian modelling with gaussian processes. In Advances in Neural Information Processing Systems 17, 2005.

65. Sebastian Thrun and Tom M Mitchell. Learning one more thing. Technical report, DTIC Document, 1994.

66. Pengcheng Wu and Thomas G Dietterich. Improving SVM accuracy by training on auxiliary data sources. In Proceedings of the 21st International Conference on Machine Learning (ICML), pages 110–117. ACM, 2004.

67. Jun Yang, Rong Yan, and Alexander G Hauptmann. Cross-domain video concept detection using adaptive svms. In Proceedings of the 15th International Conference on Multimedia, pages 188–197. ACM, 2007.

68. Yi Yao and Gianfranco Doretto. Boosting for transfer learning with multiple sources. In Proceedings of the 23rd Conference on Computer Vision and Pattern Recognition (CVPR), pages 1855–1862. IEEE, 2010.

69. K Yu, V Tresp, and A Schwaighofer. Learning gaussian processes from multiple tasks. In Proceedings of the 22nd International Conference on Machine Learning (ICML), pages 1012–1019. ACM, 2005.

70. Scikit Learn, http://scikit-learn.org/stable/

71. Baskin, I. I., Marcou, G., Horvath, D., & Varnek, A. (2017). Random Subspaces and Random Forest. Tutorials in Chemoinformatics, 263-269. doi:10.1002/9781119161110.ch18

72. Basu, S., Davidson, I., & Wagstaff, K. L. (2009). Constrained clustering: Advances in algorithms, theory, and applications. Boca Raton: CRC Press.

73. Cerrada, M., & Aguilar, J. (2008). Reinforcement Learning in System Identification. Reinforcement Learning. doi:10.5772/5273

74. Dallvechia-Adams, S., & Kuhar, M. J. (2002). CART (CART Peptides). Wiley Encyclopedia of Molecular Medicine. doi:10.1002/0471203076.emm1010

75. Dougherty, G. (2013). Pattern recognition and classification: An introduction. New York: Springer.

76. Fundamentals of Whole-System, Systemic, and Multiperspective Machine Learning. (2012). Reinforcement and Systemic Machine Learning for Decision Making, 23-56. doi:10.1002/9781118266502.ch2

77. Gollapudi, S., & Laxmikanth, V. (2016). Practical machine learning: Tackle the real-world complexities of modern machine learning with innovative and cutting-edge techniques.

78. Huang M., Niu W., & Liang X. (2009). An improved Decision Tree classification algorithm based on ID3 and the application in score analysis. 2009 Chinese Control and Decision Conference. doi:10.1109/ccdc.2009.5192865

79. Kogan, J., Nicholas, C. K., & Teboulle, M. (2006). Grouping multidimensional data: Recent advances in clustering. (Springer e-books.) Berlin: Springer.

80. Liberati, D. (n.d.). Machine Learning Through Data Mining. Machine Learning, 23-31. doi:10.4018/978-1-60960-818-7.ch103

81. Li, J., & Castagna, J. (2004). Support Vector Machine (SVM) pattern recognition to AVO classification. Geophysical Research Letters, 31(2). doi:10.1029/2003gl018299

82. Machine Learning in Healthcare Informatics. (2014). Intelligent Systems Reference Library. doi:10.1007/978-3-642-40017-9

83. MLDM (Conference), & Perner, P. (2015). Machine learning and data mining in pattern recognition: 11th International Conference, MLDM 2015, Hamburg, Germany, July 20-21, 2015, Proceedings.

84. Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). Foundations of machine learning. (Foundations of machine learning.) Cambridge, Mass. [u.a.: MIT Press.

85. The Multi-Layer Perceptron Model. (2006). A Statistical Approach to Neural Networks for Pattern Recognition, 9-18. doi:10.1002/9780470148150.ch2

86. Natarajan, B. K. (2014). Machine Learning. Elsevier Science.

87. Oladipupo, T. (2010). Introduction to Machine Learning. New Advances in Machine Learning. doi:10.5772/9394

88. Qi, X., Silvestrov, S., & Nazir, T. (2017). Data classification with support vector machine and generalized support vector machine. doi:10.1063/1.4972718

89. Wu, J. (2012). Information-Theoretic K-means for Text Clustering. Advances in K-means Clustering, 69-98. doi:10.1007/978-3-642-29807-3_4

90. Zhang Huijuan, & Sun Shixuan. (2013). A Graph Clustering algorithm based on shared neighbors and connectivity. 2013 8th International Conference on Computer Science & Education. doi:10.1109/iccse.2013.6554010

91. Scikit-learn Machine learning in Python, http://scikit-learn.org/stable/index.html

APPENDIX A

# HEART DISEASE DATASET

The description of heart disease dataset is explained in following tables:

Table A.1. List of Features and their Descriptions in the Heart Problem Data

| Feature name | Type | Description and values | % missing |
|---|---|---|---|
| Encounter ID | Numeric | Unique identifier of an encounter | 0% |
| Patient number | Numeric | Unique identifier of a patient | 0% |
| Race | Nominal | Values: Caucasian, Asian, African American, Hispanic, and other | 3% |
| Gender | Nominal | Values: male, female, and unknown/invalid | 0% |
| Age | Nominal | Grouped in 10-years intervals: [0,10), [10,20), …, [90,100) | 0% |
| Weight | Numeric | Weight in pounds | 97.5% |
| Admission type | Nominal | Integer identifier corresponding to 9 distinct values, for example, emergency, urgent, elective, newborn, and not available | 0% |
| Discharge disposition | Nominal | Integer identifier corresponding to 29 distinct values, for example, discharged to home, expired, and not available | 0% |
| Admission source | Nominal | Integer identifier corresponding to 21 distinct values, for example, physician referral, emergency room, and transfer from a hospital | 0% |
| Time in hospital | Numeric | Integer number of days between admission and discharge | 0% |
| Payer code | Nominal | Integer identifier corresponding to 23 distinct values, for example, Blue Cross\Blue Shield, Medicare, and self-pay | 43% |
| Medical specialty | Nominal | Values, for example, cardiology, internal medicine, family\general practice, and surgeon | 44.6% |

| Feature name | Type | Description and values | % missing |
|---|---|---|---|
| Number of lab Procedures | Numeric | Number of lab tests performed during the encounter | 0% |
| Number of procedures | Numeric | Number of procedures (other than lab tests) performed during the encounter | 0% |
| Number of medications | Numeric | Number of distinct generic names administered during the encounter | 0% |
| Number of outpatient visits | Numeric | Number of outpatient visits of the patient in the year preceding the encounter | 0% |
| Number of emergency visits | Numeric | Number of emergency visits of the patient in the year preceding the encounter | 0% |
| Number of inpatient visits | Numeric | Number of inpatient visits of the patient in the year preceding the encounter | 0% |
| Diagnosis 1 | Nominal | The primary diagnosis(coded as first three digits of ICD9); 848 distinct values | 0% |
| Diagnosis 2 | Nominal | secondary diagnosis(coded as first three digits of ICD9); 923 distinct values | 0.2% |
| Diagnosis 3 | Nominal | Additional secondary diagnosis(coded as first three digits of ICD9); 954 distinct values | 3% |
| Number of Diagnoses | Numeric | Number of diagnoses entered to the system | 0% |
| Glucose serum test result | Nominal | Indicates the range of the result or if the test was not taken. Values: “>200,” “>300,” “ normal,” and “none” if not measured | 0% |
| A1c test result | Nominal | Indicates the range of the result or if the test was not taken. Values: “>8” if the result was greater than 8%, “>7” if the result was greater than 7% but less than 8%, “normal” if the result was less than 7%, and “none” if not measured | 0% |
| 24 features for medications | Nominal | For the generic names : metformin, repaglinide, nateglinide, chlorpropamide, glimepiride, acetohexamide, glipizide, glyburide, tolbutamide, pioglitazone, rosiglitazone, acarbose, miglitol, troglitazone, | 0% |

| Feature name | Type | Description and values | % missing |
|---|---|---|---|
| | | tolazamide, examide, citoglipton, insulin, glyburide-metformin, glipizide-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, and metformin-pioglitazone, the feature indicates whether the drug was prescribed or there was a change in the dosage. Values: “up” if the dosage was increased during the encounter, “down” if the dosage was decreased, “steady” if the dosage did not change, and “no” if the drug was not prescribed | |
| Change of medication | | Indicates if there was a change in diabetic medications (either dosage or generic name). Values: “change” and “no change” | 0% |
| Diabetes medications | Nominal | Indicates if there was any diabetic medication prescribed. Values “yes” and “no” | 0% |
| Readmitted | Nominal | 30 day, “>30” if the patient was readmitted in more than 30 days, and “No” for no record of readmission | 0% |
| Diabetes | Nominal | Values: “1” if patient has diabetes, and “0” if patient does not have diabetes | 0% |
| Kidney Problem | Nominal | Values: “1” if patient has kidney problem, and “0” if patient does not have kidney problem | 0% |
| Ulcers, Toe, Foot, and Leg Amputation | Nominal | Values: if yes then “1”, and if no then “0” | 0% |
| Diabetes and kidney problem | Nominal | Values: if yes then “1”, and if no then “0” | 0% |
| Diabetes and ulcers, toe, foot, and leg amputation | Nominal | Values: if yes then “1”, and if no then “0” | 0% |
| Heart problem | Nominal | Values: “1” if patient has heart problem, and “0” if patient does not have heart problem | 0% |

Table A.2. Values of the Primary Diagnosis in the Used Heart Problem Data Set

| Group Name | Icd9 codes | Number of encounters | % of encounter | Description |
|---|---|---|---|---|
| Circulatory | 390-459, 785 | 5,566 | 29.03% | Disease of the circulatory system |
| Respiratory | 460-519, 786 | 3,199 | 16.69% | Disease of the respiratory system |
| Digestive | 520-579, 787 | 1,970 | 10.27% | Disease of the digestive system |
| Diabetes | 250.xx | 815 | 4.25% | Diabetes mellitus |
| Injury | 800-999 | 931 | 4.86% | Injury and poisoning |
| Musculoskeletal | 710-739 | 1,775 | 9.26% | Disease of the musculoskeletal system and connective tissue |
| Genitourinary | 580-629, 788 | 777 | 4.05% | Disease of the genitourinary system |
| Neoplasms | 140-239 | 727 | 3.79% | Neoplasms |
| | 780, 781, 784, 790-799 | 465 | 2.42% | Other symptoms, signs, and all-defined conditions |
| | 240-279, without 250 | 548 | 2.86 | Endocrine, nutritional, and metabolic diseases and immunity disorders, without diabetes |
| | 680-709, 782 | 673 | 3.5% | Diseases of the skin and subcutaneous tissue |
| | 001-139 | 249 | 1.3% | Infectious and parasitic disease |
| | 290-319 | 592 | 3.09% | Mental disorders |
| | E-V | 133 | 0.7% | External causes of injury and supplemental classification |
| other | 280-289 | 147 | 0.77% | Diseases of the blood and blood-forming organs |

| Group Name | Icd9 codes | Number of encounters | % of encounter | Description |
|---|---|---|---|---|
| | 320-359 | 205 | 1.07% | Diseases of the nervous system |
| | 630-679 | 167 | 0.87% | Complications of pregnancy, childbirth, and the puerperium |
| | 360-389 | 75 | 0.39% | Diseases of the sense organs |
| | 740-759 | 15 | 0.08% | Congenital anomalies |

Table A.3. Distribution of Variable Values and Heart Problem

| Variable | Number of encounters | %of the population | (Heart problem) Number of encounters | (Heart problem) % in group |
|---|---|---|---|---|
| Gender | | | | |
| Female | 10,497 | 54.8% | 1,165 | 11.1% |
| Male | 8,673 | 45.2% | 1,653 | 19.1% |
| Race | | | | |
| Caucasian | 13,828 | 72.1% | 2,224 | 16.1% |
| AfricanAmerican | 3,722 | 19.4% | 357 | 9.6% |
| Hispanic | 538 | 2.8% | 63 | 11.7% |
| Asian | 144 | 0.7% | 18 | 12.5% |
| Other | 371 | 2.0% | 61 | 16.4% |
| Missing | 567 | 3.0% | 95 | 16.8% |
| Medical specialty | | | | |
| Internal Medicine | 2,932 | 15.3% | 350 | 11.9% |
| Emergency/Trauma | 1,240 | 6.5% | 127 | 10.2% |
| Family/General Practice | 1,481 | 7.7% | 121 | 8.2% |
| Cardiology | 1,084 | 5.7% | 671 | 61.9% |
| Surgery | 775 | 4.0% | 38 | 4.9% |

| Variable | Number of encounters | %of the population | (Heart problem) Number of encounters | (Heart problem) % in group |
|---|---|---|---|---|
| Other | 3,104 | 16.2% | 296 | 9.5% |
| Unknown | 8,554 | 44.6% | 1215 | 14.2 |
| Glucose serum test result | | | | |
| None | 18,188 | 94.9% | 2684 | 14.8 |
| Norm | 500 | 2.6% | 64 | 12.8 |
| >200 | 268 | 1.4% | 0 | 0.0 |
| >300 | 214 | 1.1% | 0 | 0.0 |
| Admission type | | | | |
| Emergency | 9,190 | 48.0% | 1190 | 12.9 |
| Urgent | 3,343 | 17.4% | 615 | 18.4 |
| Elective | 4,429 | 23.1% | 714 | 16.1 |
| Other | 2,208 | 11.5% | 299 | 13.5 |
| Discharge disposition | | | | |
| Discharged to home | 13,503 | 70.4% | 2111 | 15.6 |
| Otherwise | 5,667 | 29.6% | 707 | 12.5 |
| Admission source | | | | |
| Admitted from emergency room | 9,836 | 51.3% | 1252 | 12.7 |
| Admitted because of physician/clinic | 6,886 | 36.0% | 1063 | 15.4 |
| referral | 2,448 | 12.8% | 503 | 20.5 |
| Otherwise | | | | |
| Age | | | | |
| 30 years old or younger | 364 | 2.0% | 1 | 0.3 |
| 30-60 years old | 7,376 | 38.5% | 1015 | 13.8 |
| Older than 60 | 11,430 | 59.6% | 1802 | 15.8 |

APPENDIX B

READMISSION DATASET

The description of readmission dataset is explained in the following tables:

Table A.4. List of Features and their Descriptions in the Readmission Data

| **Feature name** | **Type** | **Description and values** | **% missing** |
|---|---|---|---|
| Encounter ID | Numeric | Unique identifier of an Encounter | 0% |
| Patient number | Numeric | Unique identifier of a patient | 0% |
| Race | Nominal | Values: Caucasian, Asian, African American, Hispanic, and other | 2.2% |
| Gender | Nominal | Values: male, female, and unknown/invalid | 0% |
| Age | Nominal | Grouped in 10-year intervals: [0,10), [10,20), …, [90,100) | 0% |
| Admission type | Nominal | Integer identifier corresponding to 9 distinct values, for example, emergency, urgent, elective, newborn, and not available | 0% |
| Discharge disposition | Nominal | Integer identifier corresponding to 29 distinct values, for example, discharged to home, expired, and not available | 0% |
| Admission Source | Nominal | Integer identifier corresponding to 21 distinct values, for example, physician referral, emergency room, and transfer from a hospital | 0% |
| Time in hospital | Numeric | Integer number of days between admission and discharge | 0% |
| Medical Specialty | Nominal | Values, for example, cardiology, internal medicine, family\general practice, and surgeon | 0% |
| Number of lab Procedures | Numeric | Number of lab tests performed during the encounter | 0% |
| Number of procedures | Numeric | Number of procedures (other than lab tests) performed during the encounter | 0% |

| Feature name | Type | Description and values | % missing |
|---|---|---|---|
| Number of medications | Numeric | Number of distinct generic names administered during the encounter | 0% |
| Number of outpatient visits | Numeric | Number of outpatient visits of the patient in the year preceding the encounter | 0% |
| Number of emergency visits | Numeric | Number of emergency visits of the patient in the year preceding the encounter | 0% |
| Number of inpatient visits | Numeric | Number of inpatient visits of the patient in the year preceding the encounter | 0% |
| Diagnosis 1 | Nominal | The primary diagnosis(coded as first three digits of ICD9); 848 distinct values | 0% |
| Diagnosis 2 | Nominal | secondary diagnosis(coded as first three digits of ICD9); 923 distinct values | 0.3% |
| Diagnosis 3 | Nominal | Additional secondary diagnosis(coded as first three digits of ICD9); 954 distinct values | 1.4% |
| Number of Diagnoses | Numeric | Number of diagnoses entered to the system | 0% |
| Glucose serum test result | Nominal | Indicates the range of the result or if the test was not taken. Values: “>200,” “>300,” “ normal,” and “none” if not measured | 0% |
| A1c test result | Nominal | Indicates the range of the result or if the test was not taken. Values: “>8” if the result was greater than 8%, “>7” if the result was greater than 7% but less than 8%, “normal” if the result was less than 7%, and “none” if not measured | 0% |
| Change of medication | | Indicates if there was a change in diabetic medications (either dosage or generic name). Values: “change” and “no change” | 0% |
| Diabetes medications | Nominal | Indicates if there was any diabetic medication prescribed. Values “yes” and “no” | 0% |
| 24 features for medications | Nominal | For the generic names : metformin, repaglinide, nateglinide, chlorpropamide, glimepiride, acetohexamide, glipizide, glyburide, | 0% |

| Feature name | Type | Description and values | % missing |
|---|---|---|---|
| | | tolbutamide, pioglitazone, rosiglitazone, acarbose, miglitol, troglitazone, tolazamide, examide, citoglipton, insulin, glyburide-metformin, glipizide-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, and metformin-pioglitazone, the feature indicates whether the drug was prescribed or there was a change in the dosage. Values: "up" if the dosage was increased during the encounter, "down" if the dosage was decreased, "steady" if the dosage did not change, and "no" if the drug was not prescribed | |
| Readmitted | Nominal | 30 day, ">30" if the patient was readmitted in more than 30 days, and "No" for no record of readmission | 0% |

Table A.5. Values of the Primary Diagnosis in the Used Readmission Data Set

| Group Name | Icd9 codes | Number of encounters | % of encounter | Description |
|---|---|---|---|---|
| Circulatory | 390-459, 785 | 30,334 | 30.30% | Disease of the circulatory system |
| Respiratory | 460-519, 786 | 10,406 | 10.39% | Disease of the respiratory system |
| Digestive | 520-579, 787 | 9,207 | 9.2% | Disease of the digestive system |
| Diabetes | 250.xx | 8,756 | 8.75% | Diabetes mellitus |
| Injury | 800-999 | | | Injury and poisoning |
| Genitourinary | 580-629, 788 | 5,079 | 5.07% | Disease of the genitourinary system |
| Neoplasms | 140-239 | 3,432 | 3.43% | Neoplasms |

| Group Name | Icd9 codes | Number of encounters | % of encounter | Description |
|---|---|---|---|---|
| other | 240-279, without 250 | 2,701 | 2.7% | Endocrine, nutritional, and metabolic diseases and immunity disorders, without diabetes |
| | 001-139 | 2,767 | 2.76% | Infectious and parasitic disease |
| | 290-319 | 2,261 | 2.26% | Mental disorders |
| | 280-289 | | | Diseases of the blood and blood-forming organs |
| | 320-359 | 946 | 0.95% | Diseases of the nervous system |
| | 360-389 | 263 | 0.26% | Diseases of the sense organs |
| | 360-389 | 263 | 0.26% | Diseases of the sense organs |

Table A.6. Distribution of Variable Values and Readmissions

| Variable | Number of encounters | %of the population | (Readmitted) Number of encounters | (Readmitted) % in group |
|---|---|---|---|---|
| Gender | | | | |
| Female | 53,814 | 53.8% | 25,220 | 46.9% |
| Male | 46,283 | 46.24% | 20,846 | 45.0% |
| Race | | | | |
| Caucasian | 74,745 | 74.7% | 35,038 | 46.9% |
| AfricanAmerican | 18,998 | 19% | 8,679 | 45.7% |
| Hispanic | 2,015 | 2% | 843 | 41.8% |
| Asian | 630 | 0.6% | 219 | 34.8% |
| Other | 1,477 | 1.5% | 575 | 38.9% |
| Missing | 2,232 | 2.23% | 712 | 31.9% |
| Medical specialty | | | | |
| Internal Medicine | 14,422 | 14.4% | 6,248 | 43.3% |

| | | | | |
|---|---|---|---|---|
| Emergency/Trauma | 7,561 | 7.5% | 3,851 | 50.9% |
| Family/General Practice | 7,388 | 7.4% | 3,522 | 47.7% |
| Cardiology | 5,310 | 5.35 | 2,266 | 42.7% |
| Surgery-General | 3,065 | 3.1% | 1,372 | 44.8% |
| Other | 13,113 | 13% | 5,198 | 39.6% |
| Unknown | 49,238 | 49% | 23,609 | 47.9% |
| Primary diagnosis | | | | |
| Circulatory | 30,335 | 30% | 14,290 | 47.1% |
| Respiratory | 10,407 | 10.4% | 5,226 | 50.2% |
| Digestive | 9,208 | 9.2% | 4,229 | 45.9% |
| Diabetes | 8,757 | 8.75% | 4,455 | 50.9% |
| Genitourinary | 5,080 | 5% | 2,250 | 44.3% |
| Neoplasms | 3,433 | 3.4% | 1,146 | 33.4% |
| Other | 32,877 | 32.8% | 9,827 | 29.9% |
| Glucose serum test result | | | | |
| None | 94,839 | 94.75% | 43,526 | 45.9% |
| Norm | 2,541 | 2.5% | 1,147 | 45.1% |
| >200 | 1,464 | 1.5% | 0 | 0.0% |
| >300 | 1,253 | 1.3% | 0 | 0.0% |
| Admission type | | | | |
| Emergency | 53,937 | 53.89% | 25,508 | 47.3% |
| Urgent | 18,048 | 18.0% | 8,303 | 46.0% |
| Elective | 17,870 | 17.9% | 7,228 | 40.4% |
| Other | 10,242 | 10.2% | 5,027 | 49.1% |
| Discharge disposition | | | | |
| Discharged to home | 59,556 | 59.5% | 26,780 | 45.0% |
| Otherwise | 40,541 | 40.5% | 19,286 | 47.6% |
| Admission source | | | | |
| Admitted from emergency room | 57,453 | 57.4% | 28,371 | 49.4% |
| Admitted because of | 29,721 | 29.7% | 12,678 | 42.7% |
| physician/clinic referral | 12,923 | 12.9 | 5,017 | 38.8% |
| Otherwise | | | | |

| | | | | |
|---|---|---|---|---|
| Age | | | | |
| 30 years old or younger | 2,500 | 2.5% | 1,030 | 41.2% |
| 30-60 years old | 30,359 | 30.3% | 13,312 | 43.8% |
| Older than 60 | 67,238 | 67.2% | 31,724 | 47.2% |

Table A.7. Distribution of Variable Values and Heart Problem

| **Variable** | **Number of encounters** | **%of the population** | **Number of encounters** | **% in group** |
|---|---|---|---|---|
| Gender | | | | |
| Female | 10,497 | 54.8% | 1,165 | 11.1% |
| Male | 8,673 | 45.2% | 1,653 | 19.1% |
| Race | | | | |
| Caucasian | 13,828 | 72.1% | 2,224 | 16.1% |
| AfricanAmerican | 3,722 | 19.4% | 357 | 9.6% |
| Hispanic | 538 | 2.8% | 63 | 11.7% |
| Asian | 144 | 0.7% | 18 | 12.5% |
| Other | 371 | 2.0% | 61 | 16.4% |
| Missing | 567 | 3.0% | 95 | 16.8% |
| Medical specialty | | | | |
| Internal Medicine | 2,932 | 15.3% | 350 | 11.9% |
| Emergency/Trauma | 1,240 | 6.5% | 127 | 10.2% |
| Family/General Practice | 1,481 | 7.7% | 121 | 8.2% |
| Cardiology | 1,084 | 5.7% | 671 | 61.9% |
| Surgery | 775 | 4.0% | 38 | 4.9% |
| Other | 3,104 | 16.2% | 296 | 9.5% |
| Unknown | 8,554 | 44.6% | 1215 | 14.2 |
| Glucose serum test result | | | | |
| None | 18,188 | 94.9% | 2684 | 14.8 |
| Norm | 500 | 2.6% | 64 | 12.8 |
| >200 | 268 | 1.4% | 0 | 0.0 |
| >300 | 214 | 1.1% | 0 | 0.0 |
| Admission type | | | | |
| Emergency | 9,190 | 48.0% | 1190 | 12.9 |

| Variable | Number of encounters | %of the population | Number of encounters | % in group |
|---|---|---|---|---|
| Urgent | 3,343 | 17.4% | 615 | 18.4 |
| Elective | 4,429 | 23.1% | 714 | 16.1 |
| Other | 2,208 | 11.5% | 299 | 13.5 |
| Discharge disposition | | | | |
| Discharged to home | 13,503 | 70.4% | 2111 | 15.6 |
| Otherwise | 5,667 | 29.6% | 707 | 12.5 |
| Admission source | | | | |
| Admitted from emergency room | 9,836 | 51.3% | 1252 | 12.7 |
| Admitted because of physician/clinic | 6,886 | 36.0% | 1063 | 15.4 |
| referral | 2,448 | 12.8% | 503 | 20.5 |
| Otherwise | | | | |
| Age | | | | |
| 30 years old or younger | 364 | 2.0% | 1 | 0.3 |
| 30-60 years old | 7,376 | 38.5% | 1015 | 13.8 |
| Older than 60 | 11,430 | 59.6% | 1802 | 15.8 |